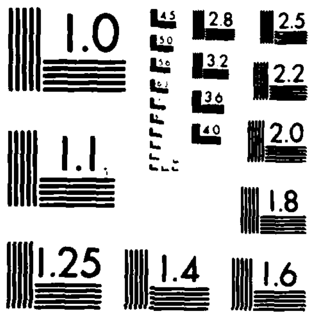
AD-A128 235 FORCE METHOD OPTIMIZATION II VOLUME II USER'S MANUAL 1/3
(U) BELL AEROSPACE TEXTRON BUFFALO NY J R BATT ET AL.
NOV 82 AFWAL-TR-82-3088-VOL-2 F33615-8-C-3214
UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

(2)

AD A128235

AFWAL-TR-82-3088
Volume II

FORCE METHOD OPTIMIZATION II
Volume II User's Manual

J. R. Batt
B. J. Dale
S. C. Skalski
D. L. Witkop
S. Gellin

Bell Aerospace Textron
Buffalo, New York 14240

November 1982

Final Report for Period August 1980 to December 1982

Approved for Public Release; Distribution Unlimited

FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Narendra S Khot

NARENDRA S. KHOT, Project Engineer
Design & Analysis Methods Group
Analysis & Optimization Branch

Fred A. Picchioni

FREDERICK A. PICCHIONI, Lt. Col.
USAF, Ch. Analysis & Optimization
Branch
Structures & Dynamics Division

For the Commander

Ralph L Kuster

RALPH L. KUSTER, JR., Col. USAF
Chief, Structures & Dynamics Div.

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/FIBRA, W-PAFB, OH 45433 to help us maintain a current mailing list".

Copies of this report should not be returend unless return is required by security considerations. contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWAL-TR-82-3088, Volume II	2. GOVT ACCESSION NO. AD-A128235	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FORCE METHOD OPTIMIZATION II, Volume II - User's Manual		5. TYPE OF REPORT & PERIOD COVERED Final Report August 1980 - December 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. R. Batt, B. J. Dale, S. C. Skalski, D. L. Witkop, S. Gellin		8. CONTRACT OR GRANT NUMBER(s) F33615-80-C3214
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bell Aerospace Textron Buffalo, New York 14240		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project 2307 Task 2307N5 Work Unit 2307N518
11. CONTROLLING OFFICE NAME AND ADDRESS Flight Dynamics Laboratory (FIBRA) AF Wright Patterson Laboratories (AFSC) Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE November 1982
		13. NUMBER OF PAGES 206
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approval for public release - distribution unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Structural Optimization
Force Method of Finite Element Analysis
Minimum Weight Design
Rapid Reanalysis and Vulnerability Analysis

This docum

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Investigates the utilization of the force method of finite element analysis for the automatic iterative design of aircraft structures with stress, displacements, maximum and minimum size and dynamic constraints. Developes a rapid reanalysis method based on the force method for damage assessment. Research has resulted in a computer code named OPTFORCE II an expansion of code OPTFORCE I documentated in AFWAL-TR-80-3006. Multiple loading capabilities and four finite elements have been included. These are: membrane triangle, membrane quadrilateral, shear panel and bar (axial force). Examples of —

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

BLOCK 20 (continued)

problems solved by the OPTFORCE II code are presented and compared to the optimization code OPTIM III for purposes of establishing the efficiency of the "force" method vs. the "displacement" method of analysis. A technical discussion of the research conducted is presented wherein conclusions and recommendations for future research topics are given.

FOREWORD

This report describes the work performed by Bell Aerospace Textron, a Division of Textron, Inc. Buffalo, New York. The work was sponsored by the Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio, under contract F33615-80-C3214.

The work was initiated under Project 2307, "Research in Flight Vehicle Dynamics", Task 2307N518, "Basic Research in Structure and Dynamics". The work was administered by Dr. N. S. Khot, Project Engineer of the Structures & Dynamics Division (FIBRA).

The contracted work was performed between August 1980 and December 1982.

The work was performed in the Structure and Vehicle Systems Directorate, Bell Aerospace Textron. Mr. James R. Batt was the Program Manager/Technical Director of the study.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1.0 INTRODUCTION

This volume of the report discusses the computer program OPTFORCE II from the User's standpoint. Volume I, Ref. 1, discusses the theoretical development of force method optimization, as well as the governing equations and solution procedure logic incorporated into the OPTFORCE II code. The reader must refer to Ref. 1 for in-depth discussions of these items.

Included in the OPTFORCE II program is the capability for analyzing (static and dynamic) and weight optimizing truss and aerospace type structures. Both isotropic and orthotropic material behavior can be considered. Four finite elements have been included; namely, truss (rod), symmetric shear web, triangular and quadrilateral shaped membranes. Only the membrane elements are capable of handling composite structures by utilizing a layering procedure. Section II discusses the aforementioned items and in addition an overview of program capabilities and options is presented. The weight optimization procedure is also briefly described in that section.

Description of the OPTFORCE II program itself is given in Section 3.0. Particular emphasis has been given to those sub-sections which describe the input and output data. It is noted here that all input data cards are compatible with NASTRAN formats. Each required input card is discussed thoroughly and illustrated in the sample problem solution. Output options are described and typical output format is given with the sample problem.

A programmer's manual is given in Appendix A of this volume. It's purpose is to fully describe program logic and the required peripheral storage. Individual sub-routine write-ups are presented. A short description of each routine is included to aid the reader's understanding of the program. This manual is specifically written for the computer programmer, however.

The rapid reanalysis development has resulted in a computer program described in Appendix B. Program capabilities, limitations, input and output data descriptions and illustrative examples are given. The detailed theoretical development of the rapid reanalysis procedure is presented in Volume I of this report, Reference 1.

2.0 OPTFORCE II OVERVIEW

The governing equations and solution procedure logic for the force-method weight optimization code, OPTFORCE II, is developed "in-toto" in Volume I, Ref. 1, of this report. Because of the vast amount of technical material presented in that volume only a brief description of program capabilities, finite elements and the weight optimization procedure is given.

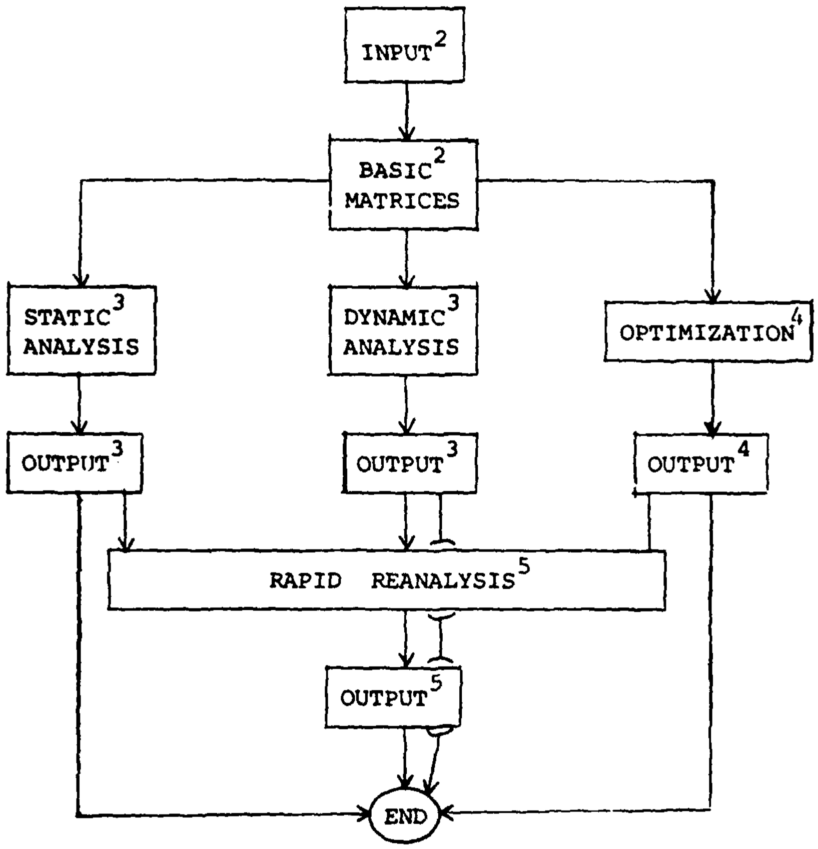
The main objective of the work reported herein and in Volume I was to develop a method of weight optimization using the force method of structural analysis. As work progressed it became apparent that additional analysis capabilities other than weight optimization were readily available and these should be made accessible to the structural analyst. As a result, options in OPTFORCE II are available to conduct several types of analyses. These are shown in Figure 1 in flow chart form. Each of the analysis blocks are further expanded in Figures 2 to 5. Examination of these figures shows the scope of analyses available. All of these are resident in OPTFORCE II except the rapid reanalysis capability. This analysis tool is described in Appendix B of this report. The matrices noted in the flow charts are defined in Section 2.0 of Volume 1, however a few definitions are given to clarify the interpretation of the flow charts.

The matrix [A] relates the external grid point forces {P} to internal stress state {S} and load reactions {R}. [f] is a normalized diagonal flexibility matrix for the entire structure; $[\bar{\Phi}]$, $[\bar{\Psi}]$ & $[\bar{\Omega}]$ are defined as:

$$[\bar{\Phi}] = [b_1]^T [\bar{f}] [b_1] \quad (1)$$

$$[\bar{\Psi}] = [b_1]^T [\bar{f}] [D] \quad (2)$$

$$[\bar{\Omega}] = [D]^T [\bar{f}] [D] \quad (3)$$



The superscripts refer to other flow charts for more information.

Figure 1 General Program Outline

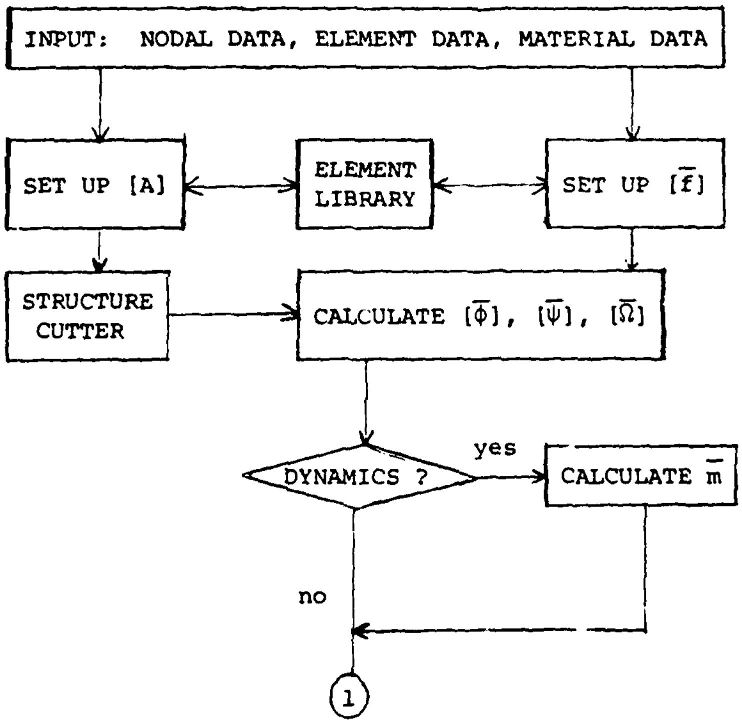


Figure 2 Basic Matrices

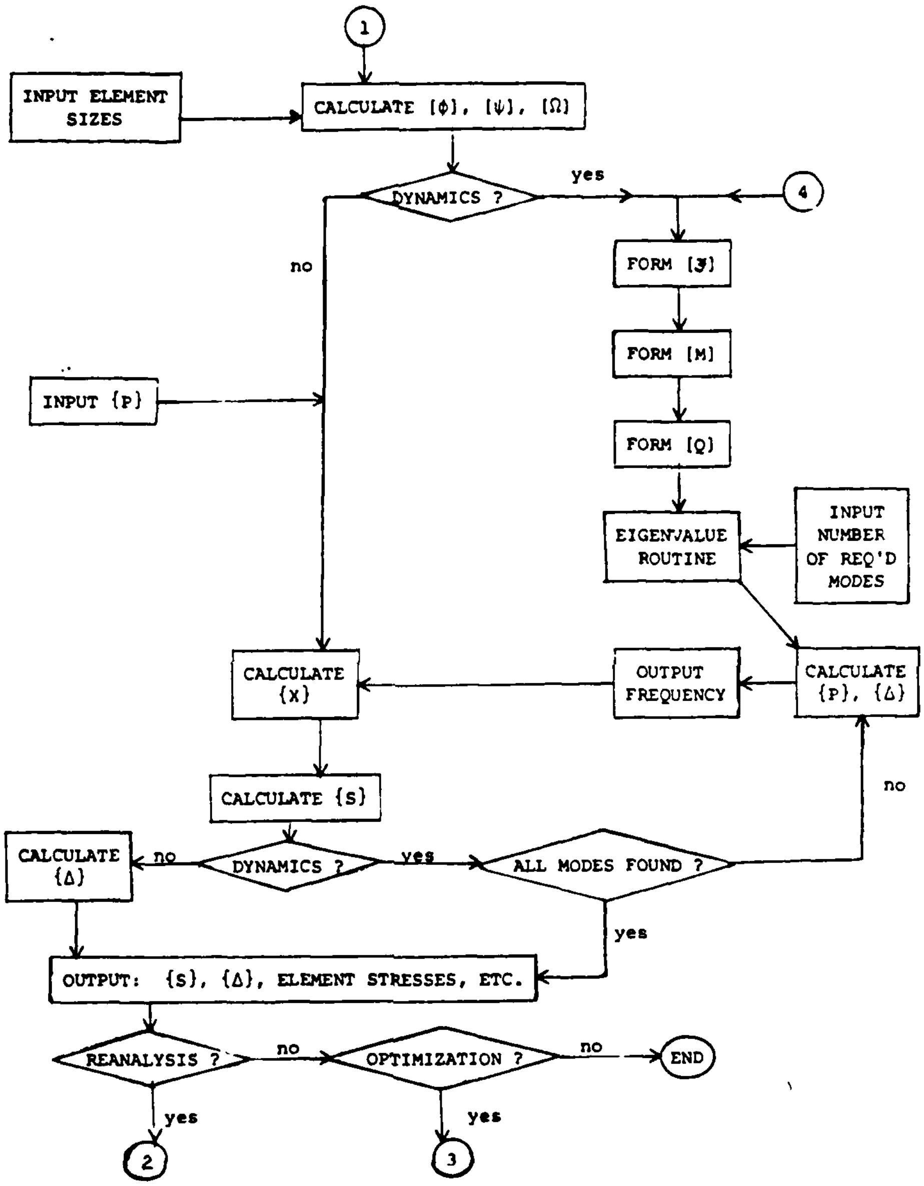


Figure 3 Basic Analyses

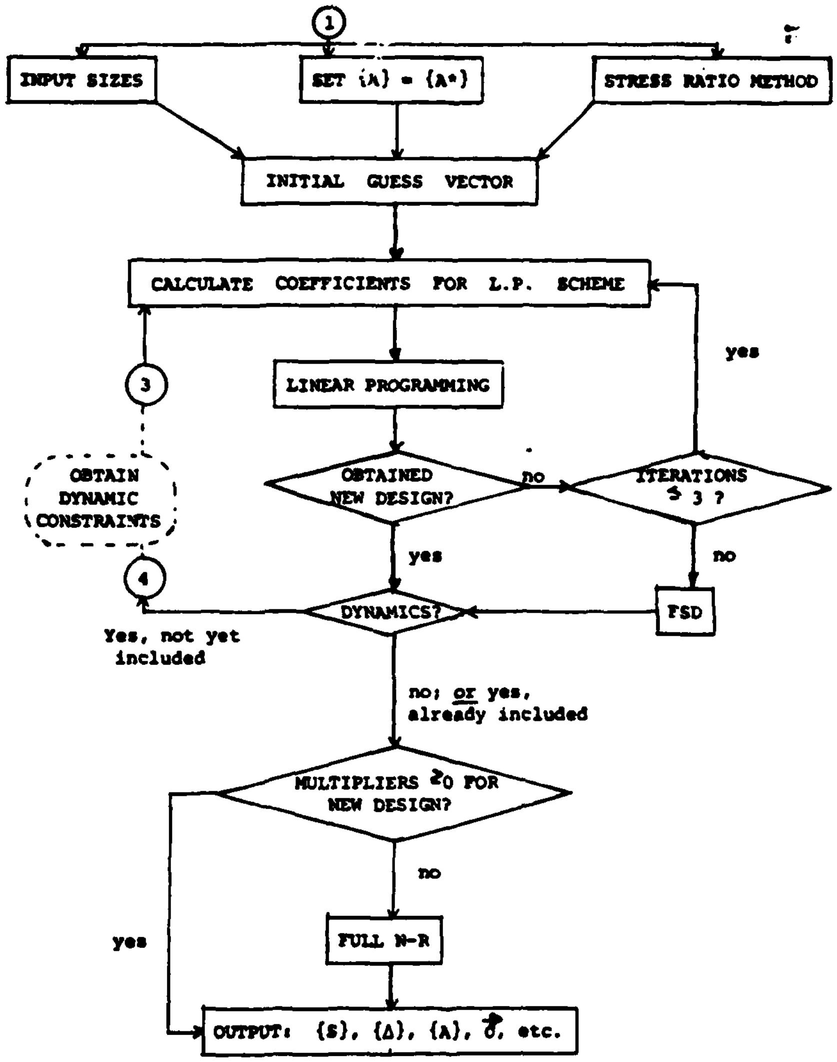


Figure 4 Optimization

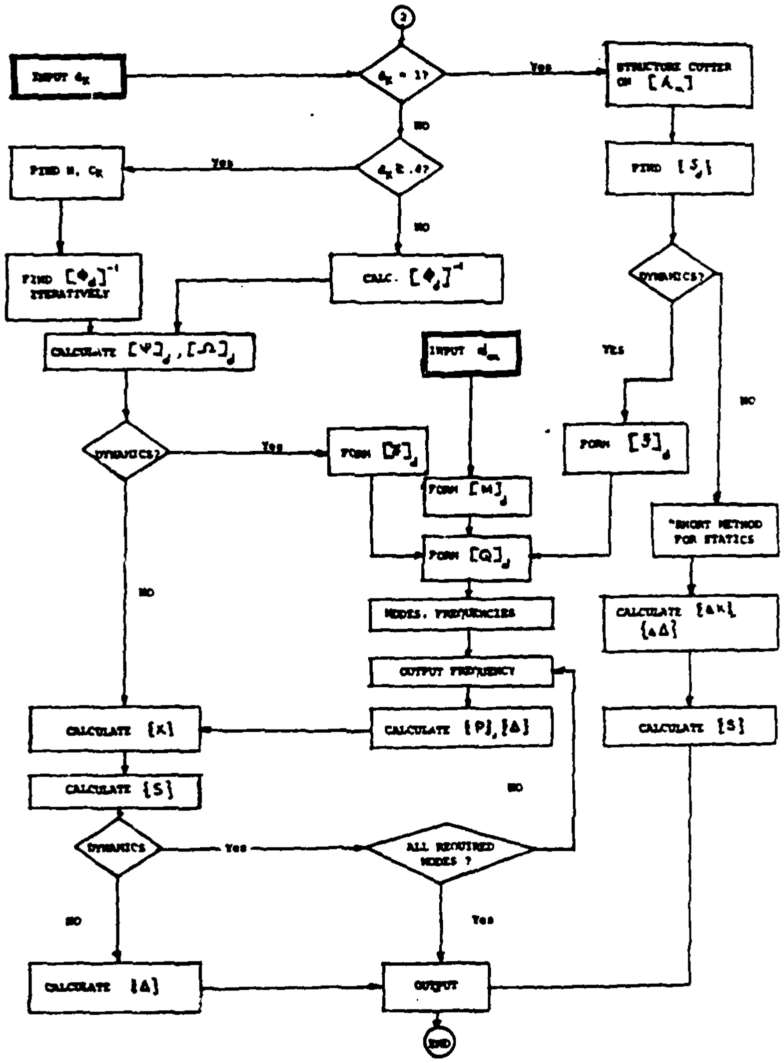


Figure 5 Rapid Reanalysis

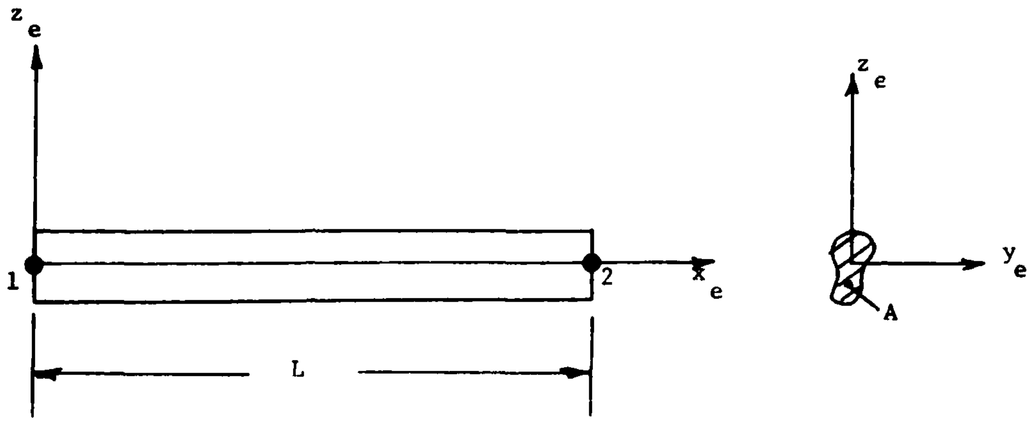
$[b_1]$ & $[D]$ arise from applying a structure cutter procedure; where $[b_1]$ represents unit values of self-equilibrating force systems and $[D]$ represents forces which are in static equilibrium with the external forces $\{P\}$. The global flexibility matrix is defined by $[F]$ which relates $\{P\}$ to $\{\Delta\}$,$\{\Delta\}$ being the displacement of the structure. $[M]$ is, of course, the lumped mass matrix of the structure. The set of redundants is defined by the vector $\{X\}$ and is a basic unknown in the force method as described in References 1 and 2.

The weight optimization procedure coded and described in detail in Volume 1, Section 2.3.3 is a modification and expansion of the one developed in Ref. 2. Only the salient features of the procedure are given here. Reference to Figure 4 will aid in the cursory description of the solution algorithm. The initial guess vector for structural member sizes (design variables) can be obtained in one of four ways. First, the initial sizes may be obtained directly from input data (OPDVIR cards) or they may be set to the minimum sizes A*. Each of these may be further used in a stress ratio method. These four options are controlled through input codes on the OPTIM card. Selection of the input guess vector permits entry into the linear programming (L.P.) scheme which solves for sets of Lagrange multipliers μ. The particular set of μ's solved for imply satisfaction of the constraints associated with the non-zero multipliers. The constraints considered in OPTFORCE II are: minimum sizes of design variables, member stress, structural displacement and natural frequency of vibration. The linear programming phase is cycled three times as shown in Figure 2.4 to obtain a new design. If a new design is not converged upon within

three iterations the program exits L.P. and produces a full stressed design (FSD). Note that the frequency constraint, if desired, is imposed at this time and the linear programming phase is re-entered. The Lagrange multipliers are now checked for non-negativity and the remaining constraints are checked for violations. If all the μ's are positive and no constraint is violated, the minimum weight solution is found as the FSD or the design implied by the LP scheme had it converged. If a particular μ is negative then the corresponding constraint is "turned-off" and the multiplier is set equal to zero. If a constraint is violated the corresponding sizes of design variables are changed to negate this violation. The current design at this point in the solution procedure is expressed as a system of unknowns in element sizes, structural redundants and the non-zero Lagrange multipliers. This system is non-linear in nature and is solved using the Newton-Raphson procedure. Output from the optimization procedure is, of course, minimum structural weight, member stresses, values of the design variables and the structures displacement behavior.

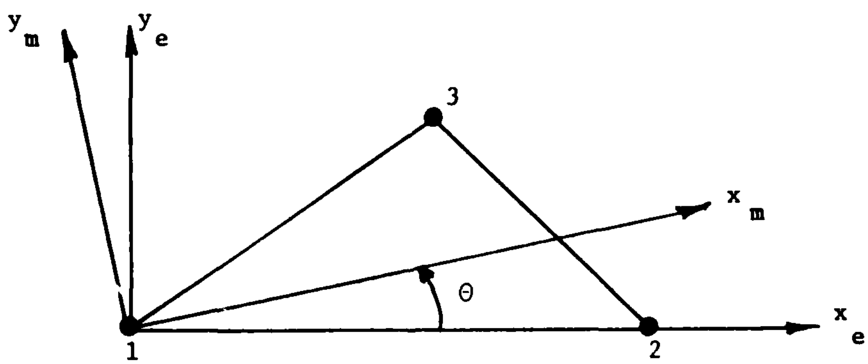
Five finite elements are provided in OPTFORCE II; namely, truss or rod, plane stress triangle, plane stress quadrilateral and two shear panels. The truss elements permit analysis of truss type structures like transmission towers for example and more importantly portions of aerospace structures such as lifting surfaces. The symmetric shear panel is of direct use here since it allows modeling of only one-half of the lifting surface. The plane stress elements are membrane type elements and consequently only accept in-plane loadings. Figure 6 displays the above elements and their characteristics. Notable among these are the design variables (denoted as the vector $\{A\}$).

Note that the membrane elements can be used to model composite materials.



Element characteristics: Two grid points, length & design variable cross-sectional area A. Isotropic materials only. Stress output σx.

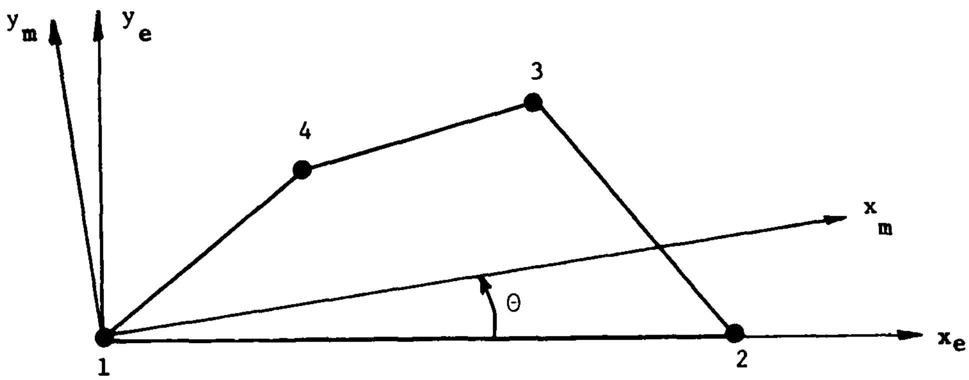
a) Truss (rod) Axial Force Member ~ ROD



Element Characteristics: Three grid points, surface area As, design variable plate thickness t, angle of orthotropy Θ. Orthotropic and isotropic materials. Centroidal stress output σx_m, σy_m, τxy_m.

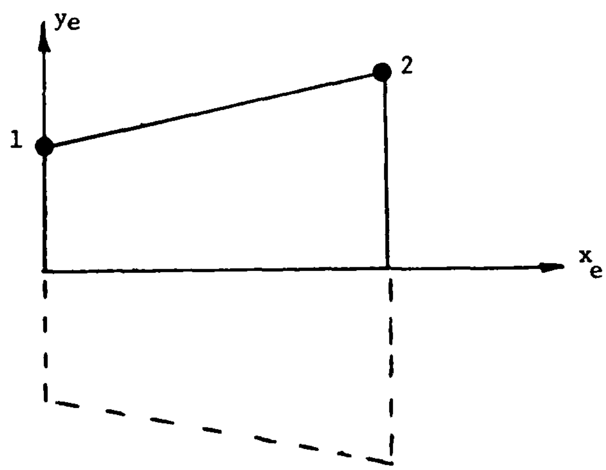
b) Triangle Membrane Plate ~ TRMEM

Figure 6 Finite Element Library



Element Characteristics: Four grid points, surface area As, design variable plate thickness t, angle of orthotropy Θ. Orthotropic and isotropic materials. Centroidal stress output σ_{x_m}, σ_{y_m}, τ_{xy_m}.

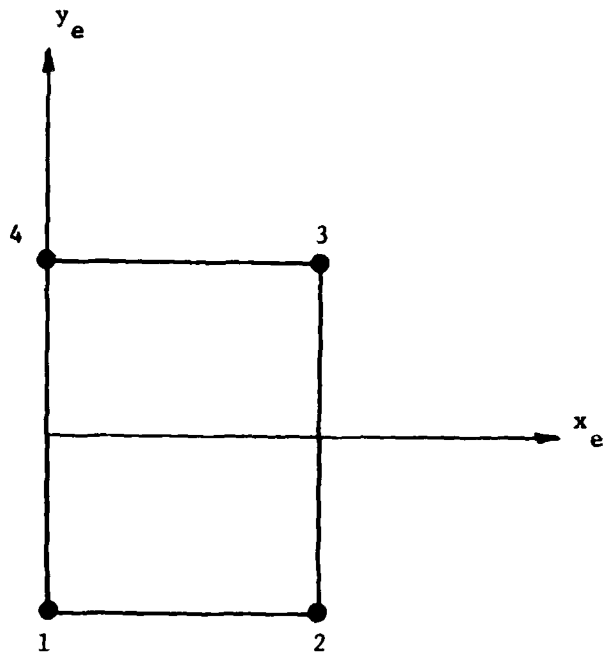
c) Quadrilaterial Membrane Plate ~ QDMEM1



Element Characteristics: Two grid points, surface area As, design variable web thickness t. Isotropic materials only. Centroidal stress output τxy. N.B. always define grid points in positive global Z direction.

d) Symmetric Shear Panel ~ WEB

Figure 6 (cont'd).



Element Characteristics: Four grid points, surface area As, design variable web thickness t. Isotropic materials only. Centroidal stress output τ_{xy}. N.B. Always define grid points in positive global Z direction.

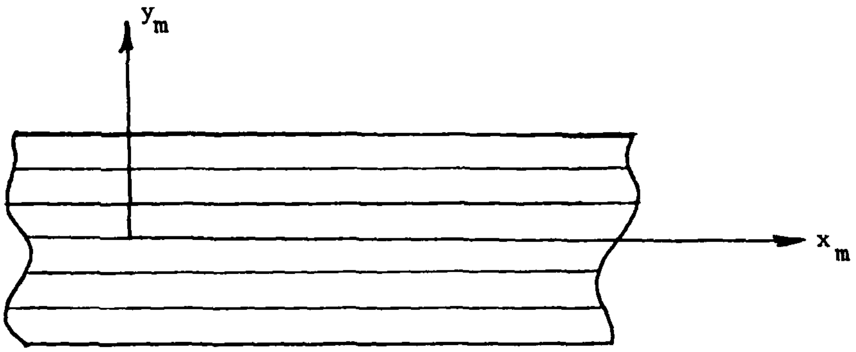
e) Shear Web ~ SHEAR

Figure 6 (cont'd.)

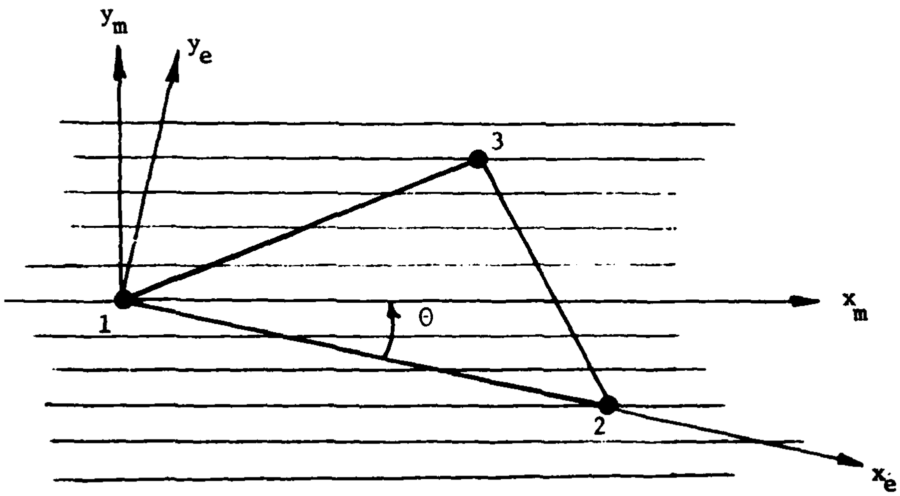
This was accomplished by providing an element layering capability coupled with orthotropic material behavior. Each layer of the plate is idealized as a separate membrane plate finite element. When modeling a composite plate consisting of many layers, the gridpoints of all the membrane elements or layers of the plate are selected to be the same. This restriction ensures that each of the layers of the composite plate are subjected to the same deformation state and is in conformance with small displacement plate theory in which it is assumed that straight normals to the plate median surface before deformation remain straight after deformation. To ensure that this condition is satisfied, the layers of the composite plate must be oriented symmetrically with respect to the median surface of the composite plate.

Each layer in a composite membrane plate was assumed to be composed of fibers embedded in a matrix material. The fibers are assumed to be oriented in such a way as to result in their characterization as an orthotropic material. A typical orthotropic layer of this type is illustrated in Figure 7. The coordinate system (x_m, y_m) are the material axes of orthotropy in which the x_m axis is oriented parallel to the fibers. The angle θ gives the orientation of the material axis with respect to the local element x_e axis. This axis is defined by the side of the element connecting grid points 1 and 2. Note that θ is measured in a counterclockwise direction, from the x_e axis to the x_m axis.

The relationship between the elastic stresses and strains for an orthotropic material in plane stress referenced to the material axis of orthotropy shown in Figure 7 is given by:



a) Typical Fiber-reinforced Layer



b) Typical Orientation of Membrane Element in Fiber-reinforced Composite Layer

Figure 7 Fiber-reinforced Composite Idealization

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix}_m = \begin{bmatrix} G_{11}, & G_{12}, & G_{13} \\ G_{21}, & G_{22}, & G_{23} \\ G_{31}, & G_{32}, & G_{33} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{bmatrix}_m \tag{4}$$

where: $G_{11} = \frac{E_x}{\bar{\nu}}$, $G_{22} = \frac{E_y}{\bar{\nu}}$, $G_{12} = \frac{\nu_{xy} E_y}{\bar{\nu}}$

$G_{21} = G_{12}$, $G_{13} = G_{31} = G_{23} = G_{32} = 0.0$

$G_{33} = G_{xy}$, $\bar{\nu} = 1 - \nu_{xy}\nu_{yx}$, $\frac{E_x}{E_y} = \frac{\nu_{xy}}{\nu_{yx}}$

These stress-strain relations are used in the calculation of membrane finite element characteristics and furthermore are reducible to isotropic material behavior. As a result it is possible to construct a composite plate composed of both fiber-reinforced orthotropic materials and isotropic metallic-type materials. It is noted here that the MAT1 and MAT2 input data cards are used to define the above material properties.

The Mises-Henchy stress failure criteria has been inserted into the OPTFORCE II program as follows:

$$Y_i = A_i \left(\sigma_x^{\,2} + \sigma_y^{\,2} - \sigma_x\sigma_y + 3\tau_{xy}^{\,2}\right)^{\frac{1}{2}} \tag{5}$$

where σ_x, σ_y and τ_{xy} are element stresses calculated in the local element x_m-y_m axis system. A_i is the element's design variable; for a bar element, for example, A_i would be its cross-sectional area. The above criteria was used to formulate the stress constraint expression:

$$g_\sigma^i = \frac{Yi}{A_i\sigma_i{}^*} - 1 \leq 0 \tag{6}$$

where $\sigma_i{}^*$ is the yield stress or some other failure stress value for the $i^{\underline{th}}$ finite element. It is a User input and is given on the MAT1, MAT2 cards. The quantity $\frac{Yi}{Ai}$ is output with the local stresses for each element.

3.0 OPTFORCE II COMPUTER PROGRAM DESCRIPTION

3.1 Input Overview

This section describes the input data required to execute the NASTRAN compatible optimization program OPTFORCE II. This input consists of a deck beginning with "BEGIN BULK" and ending with "ENDDATA". All data cards are optional except those defining controls, loads, grid, and boundary conditions. Elements must also be defined.

The NASTRAN input feature of submitting data in any order, either left or right adjusted, has been preserved. The GRID cards, SPC cards, FORCE, and MAT1, MAT2 cards are similar to NASTRAN input. The following is a summary of OPTFORCE II Input Cards:

BEGIN Bulk	First card of input
CONROD	Axial force member property and connection
CQDMEM1	Quadrilateral membrane connection
CROD	Axial force element connection
CSHEAR	Shear panel element connection
CTRMEM	Triangular membrane element connection
CWEB	Shear web element connection
FORCE	Static load at grid point
GRID	Grid point coordinates
*ICON	Displacement constraints
MAT1	Isotropic material properties
MAT2	Anisotropic material properties
*OPDVIR	Selected element design variables
OPLOADS	Loads for optimization
OPTIM	Optimization control parameters
PQDMEM1	Property card for quadrilateral membrane
PROD	Property card for axial force member
PSHEAR	Property card for shear panel
PTRMEM	Property card for triangular membrane
PWEB	Property card for shear web
SPC	Single point constraint
SPC1	Sets of single point constraints
TITLE	Title card information
ENDDATA	End of data deck

3.2 Description of Input Data

In this section, detailed OPTFORCE II input data card descriptions are presented. Each data card is described individually in alphabetical order.

---

*These cards are optional

Input Data Card BEGIN Bulk

Description: First Card of Input

1	2	3	4	5	6	7	8	9	10
BEGIN Bulk									

Input Data Card CØNRØD Axial Force Element Property and Connection

Description: Defines an axial force element (RØD) without reference to a property card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CØNRØD	EID	G1	G2	MID	A				
CØNRØD	2	16	17	23	2.69				

Field	Contents
EID	Unique element identification number (Integer >0)
G1, G2	Grid point identification numbers of connection points (Integer >0; G1 ≠ G2)
MID	Material identification number (Integer >0)
A	Area of member (Real)

Remarks:

1. Element identification numbers must be unique with respect to all other element identification numbers.
2. CØNRØD cards may only reference MAT1 material cards.
3. A is used as the design variable minimum for optimization runs, or as the design variable for statics ordynamic runs. If an OPDVIR card group is included in the bulk data deck then A is ignored.
4. See CRØD for alternative method of rod definition.

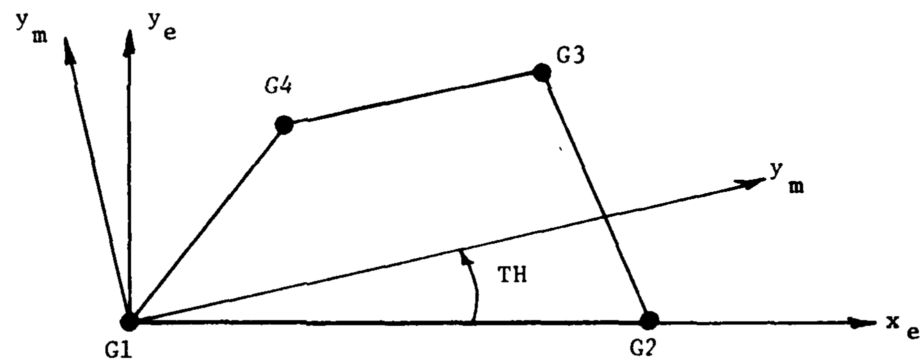
Input Data Card CQDMEM1 Quadrilateral Element Connection

Description: Defines a quadrilateral membrane element (QDMEM1).

Format and Example:

1	2	3	4	5	6	7	8	9	10
CQDMEM1	EID	PID	G1	G2	G3	G4	TH		
CQDMEM1	72	10	13	14	15	16	29.2		

Field	Contents
EID	Element identification number (Integer >0)
PID	Identification number of a PQDMEM1 property card (Integer >0)
G1, G2, G3, G4	Grid point identification numbers of connection points (Integer >0); G1 ≠ G2 ≠ G3 ≠ G4)
TH	Material property orientation angle in degrees (Real). The sketch below gives the sign convention for TH.



Remarks:

1. Element identification numbers must be unique with respect to all other element identification numbers.

2. Gridpoints G1 through G4 must be ordered consecutively around the perimeter of the element.

3. All interior angles must be less than 180 degrees.

Input Data Card CRØD Axial Force Element Connection

Description: Defines an axial force element (RØD) with reference to a property card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CRØD	EID	PID	G1	G2	EID	PID	G1	G2	
CRØD	12	13	21	23	3	12	24	5	

Field	Contents
EID	Element identification number (Integer >0)
PID	Identification number of a PRØD property card
G1, G2	Grid point identification numbers of connection points (Integer >0; G1 $\neq$ G2)

Remarks:

1. Element identification numbers must be unique with respect to all other element identification numbers.
2. See CØNRØD for alternative method of rod definition
3. One or two RØD elements may be defined on a single card.

Input Data Card CSHEAR Shear Panel Element Connection

Description: Defines a shear panel element (SHEAR).

Format and Example:

1	2	3	4	5	6	7	8	9	10
CSHEAR	EID	PID	G1	G2	G3	G4			
CSHEAR	3	6	1	5	3	7			

Field	Contents
EID	Element identification number (Integer >0)
PID	Identification number of a PSHEAR property card
G1, G2 G3, G4	Grid point identification numbers of connection points (Integer >0; $G1 \neq G2 \neq G3 \neq G4$)

Remarks:

1. Element identification numbers must be unique with respect to all other element identification numbers.
2. Grid points G1 thru G4 must be ordered consecutively around the perimeter of the element.

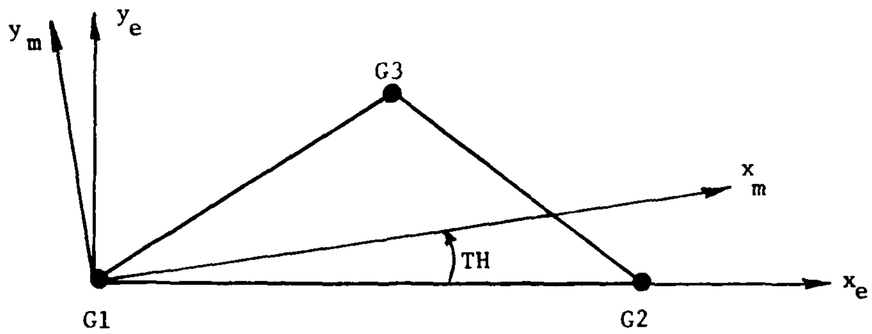
Input Data Card CTRMEM Triangular Element Connection

Description: Defines a triangular membrane element (TRMEM).

Format and Example:

1	2	3	4	5	6	7	8	9	10
CTRMEM	EID	PID	G1	G2	G3	TH			
CTRMEM	16	2	12	1	3	16.3			

Field	Contents
EID	Element identification number (Integer >0)
PID	Identification number of a PTRMEM property card
G1, G2, G3	Gridpoint identification numbers of connection points (Integer >0; G1 $\neq$ G2 $\neq$ G3)
TH	Material property orientation angle in degrees (Real). The sketch below gives the sign convention for TH.



Remarks: 1. Element identification numbers must be unique with respect to all other element identification numbers.

Input Data Card CWEB Shear Web Element Connection

Description: Defines a 2 node symmetric web element (Web) of the structural model.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CWEB	EID	PID	G1	G2					
CWEB	4	2	1	3					

Field	Contents
EID	Element identification number
PID	Identification of PWEB property card
G1, G2	Gridpoint identification numbers

Remarks: 1. See Figure 2.6

Input Data Card FØRCE Static Load

Description: Defines a static load at a grid point by specifying a vector.

Format and Example:

1	2	3	4	5	6	7	8	9	10
FØRCE	SID	G		F	N1	N2	N3		
FØRCE	2	5		2.9	0.0	1.0	0.0		

Field	Contents
SID	Load set identification number (Integer >0)
G	Gridpoint identification number (Integer >0)
F	Scale factor (Real)
N1, N2 N3	Components of Vector (Real)

Remarks:

1. The static load applied to gridpoint G is given by

$$\vec{f} = F\,\vec{N}$$

where $\vec{N}$ is the vector defined in fields 6, 7, and 8.

2. Load set is selected on the OPLOADS card.

Input Data Card GRID Grid Point

Description: Defines the location of a geometric gridpoint of the structural model and its permanent single-point constraints.

Format and Example:

1	2	3	4	5	6	7	8	9	10
GRID	ID		X1	X2	X3		PS		
GRID	2		1.0	2.0	3.0		13		

Field	Contents
ID	Gridpoint identification number (0< Integer< 999999)
X1, X2 X3	Location of the grid point
PS	Permanent single-point constraints associated with gridpoint (any of the digits 1-3 with no imbedded blanks) (Integer ≥0 or blank)

Remarks:

1. The coordinate system defined on all GRID cards is called the Global Coordinate System. All degress-of-freedom, constraints, and solution vectors are expressed in the Global Coordinate System.

Input Data Card ICON Displacement Constraints

Description: Defines the displacement constraint limits (both lower and upper) and gridpoint components which require constraints.

Format and Example:

1	2	3	4	5	6	7	8	9	10
ICON	Components	Lower	Upper	G1	G2	G3	G4	G5	
ICON	13	1.0	3.0	1	2	3	6		

Field	Contents
Components	1-3 signifies component numbers
Lower	Lower limit
Upper	Upper limit
G1, G2 G3, etc.	Gridpoint numbers. Up to 5 gridpoint numbers may appear on one ICON card.

Remarks:

1. Each ICON card contains uo to 5 gridpoints. When more than 5 are desired, list them on separate ICON cards.

2. For example, let there be the following individual constraints: U1, U2, W1, W2 greater than 1.0 and less than 4.0. V1, V2, V3 greater than 1.0 and less than 8.0.

   The required ICON cards would be:

1	2	3	4	5	6	7
ICON	13	1.0	4.0	1	2	
ICON	2	1.0	8.0	1	2	3

Input Data Card MAT1 Material Property Definition

Description: Defines the material properties for linear, temperature-independent, isotropic materials.

Format and Example: (Consists of two cards)

1	2	3	4	5	6	7	8	9	10
MAT1	MID	E	G	NU	RHØ				+abc
MAT1	17	3.+7	1.9+7		4.28				ABC

+abc	SL	SU
+ABC	20.+4	15.+4

Field	Contents
MID	Material identification number (Integer >0)
E	Young's modulus (Real ≥0.0 or blank)
G	Shear modulus (Real ≥0.0 or blank)
NU	Poisson's ratio (-1.0 <Real ≤0.5 or blank)
RHO	Material density (Real) (lbs/cu. in.) or (lbs/cu. ft.)
SL,SU	Lower stress limit, upper stress limit

Remarks:

1. One of E or G must be positive (i.e. either E >0.0 or G >0.0 or both E and G may be >0.0).

2. If any one of E, G or NU is blank, it will be computed to satisfy the identify E = 2(1+NU)G; otherwise, values supplied by the user will be used.

3. The material identification number must be unique for all MAT1 and MAT2 cards.

4. SU is used as the upper limit for the Mises-Hencky criteria of the element referencing this material during an optimization analysis.

Input Data Card MAT2 Material Property Definition

Description: Defines the material properties for linear, anisotropic materials.

Format and Example: (consists of 2 cards)

1	2	3	4	5	6	7	8	9	10
MAT2	MID	G11	G12	G13	G22	G23	G33	RHO	+abc
MAT2	13	6.2+3			6.2+3		5.1+3	0.056	BC
+abc						SL	SU		
+BC						20.+5	15.+3		

Field	Contents
MID	Material identification number (Integer >0)
Gij	The material property matrix (Real)
RHO	Material density (Real) (lbs/cu.in.) or (lbs/cu. ft.)
SL, SU	Lower stress limit, upper stress limit

Remarks:

1. The material identification numbers must be unique for all MAT1, MAT2 cards.

2. The convention for the Gij in fields 3 through 8 is represented by the matrix relationship.

$$\begin{Bmatrix} \sigma 1 \\ \sigma 2 \\ \tau 12 \end{Bmatrix} = \begin{bmatrix} G11 & G12 & G13 \\ G12 & G22 & G23 \\ G13 & G23 & G33 \end{bmatrix} \begin{Bmatrix} \varepsilon 1 \\ \varepsilon 2 \\ \gamma 12 \end{Bmatrix}$$

3. Only TRMEM and CQDMEM elements may use MAT2 cards.

Input Data Card OPDVIR

Description: Defines design variables for selected elements. These design variables are used as starting guesses in an optimization run or else as specified values in a statics or dynamics run.

Format and Example:

1	2	3	4	5	6	7	8	9	10
OPDVIR	E1	$DVAR_1$	E2	$DVAR_2$	E3	$DVAR_3$	E4	$DVAR_4$	
OPDVIR	1	.1	3	.6	4	.50	7	.23	

Field	Contents
E1, E2, etc.	Element numbers for which design variables are input.
$DAVR_1$, $DVAR_2$, etc.	Design variable starting guesses for elements 1, 2, etc.

Remarks:

1. Up to 4 elements may be defined on one OPDVIR card. More elements may be defined by successive OPDVIR cards.

2. The use of these values is dependent on the control option selected on the OPTIM input card - field #9, OPT.

   a. If OPT = GOPT the program will optimize using the design variable minimums input in this section as a starting guess.

   b. If OPT = GSTA, the program will perform a statics solution using the design variables input in this section.

   c. If OPT = GDYN, the program will perform a dynamics solution using the design variables input in this section.

   d. If OPT = GOPD, the program will optimize with dynamic constraints using the design variables input in this section.

Input Data Card OPLOADS

Description: Defines the character of the loads used in the optimization procedure.

Format and Example:

1	2	3	4	5	6	7	8	9	10
OPLOADS	LOAD			F_i					
OPLOADS	1			5					

Field	Contents
LOAD	LOAD number identification (any number)
F_i	SID number on FORCE card(s)

Remarks:

1. Each FORCE card defining this external load must contain the SID number entered for F_i.

2. Each external load requires an OPLOADS entry.

Input Data Card OPTIM

Description: Defines various control parameters for the OPTFORCE II program.

Format and Example:

1	2	3	4	5	6	7	8	9	10
OPTIM	PRI	PRS	ISTRES	NMODE	MAX	CONVD	WS	OPT	
OPTIM	No	Yes	Yes	6	50	.0005	495.0	OPT	

Field		Contents
PRI	No/Yes	Print intermediate debug information (file IO1)
PRS	No/Yes	Print stresses and displacements every iteration
ISTRES	No/Yes	Use stress-ratio method for initial guess when optimizing
NMODE		Number of modes in dynamic analysis
MAX		Maximum number of iterations permitted (default=50)
CONVD		Convergence criteria (default=.0001)
WS		Frequency constraint limit (in Hertz)
OPT	OPT	Optimize using design variable on property card
	GOPT	Optimize using OPDVIR starting guess
	STA	Statics analysis using design variable on property card
	GSTA	Statics analysis using OPDVIR starting guess
	DYN	Dynamics analysis using design variable on property card
	GDYN	Dynamics analysis using OPDVIR starting guess
	OPTD	Optimize with dynamic constraint using design variable on property card
	GOPD	Optimize with dynamic constraint using OPDVIR starting guess

Remarks: CONVD - Convergence Criteria (design variables, eigen vectors, redundants and Lagrange multipliers). If two successive iterations of all variables meet the criteria $\left|\frac{An-An+1}{An}\right| \leq CONVD$ then the iteration process is terminated.

Input Data Card PQDMEM1 Quadrilateral Membrane Property

Description: Defines the properties of the quadrilateral membrane as referenced by the CQDMEM1 card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PQDMEM1	PID	MID	T		PID	MID	T		
PQDMEM1	235	2	0.5						

Field	Contents
PID	Property identification number (Integer >0)
MID	Material identification number (Integer >0)
T	Minimum thickness of membrane (Real >0.0)

Remarks:

1. All PQDMEM1 cards must have unique property identification numbers.
2. One or two quadrilateral membrane properties may be defined on a single card.

Input Data Card PRØD Rod Property

Description: Defines the properties of a rod as referenced by the CRØD card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PRØD	PID	MID	A						
PRØD	17	23	42.6						

Field	Contents
PID	Property identification number (Integer >0)
MID	Material identification number (Integer >0)
A	Area of rod (Real)

Remarks:

1. PRØD cards must all have unique property identification numbers.
2. PRØD cards may only reference MAT1 material cards.
3. For remarks on use of A, see CONROD card description.

Input Data Card PSHEAR Shear Panel Property

Description: Defines the elastic properties of a shear panel as referenced by the CSHEAR card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PSHEAR	PID	MID	T		PID	MID	T		
PSHEAR			4.9		14	6	4.9		

Field	Contents
PID	Property identification number (Integer >0)
MID	Material identification number (Integer >0)
T	Minimum thickness of shear panel (Real $\neq$ 0.0)

Remarks:
1. All PSHEAR cards must have unique identification numbers.
2. PSHEAR cards may only reference MAT1 material cards.
3. One or two shear panel properties may be defined on a single card.

Input Data Card PTRMEM Triangular Membrane Property

Description: Defines the properties of a triangular membrane element as referenced by the CTRMEM card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PTRMEM	PID	MID	T		PID	MID	T		
PTRMEM	17	23	4.25						

Field	Contents
PID	Propertv identification number (Integer >0)
MID	Material identification number (Integer >0)
T	Membrane thickness (Real >0.0)

Remarks:

1. All PTRMEM cards must have unique property identification numbers.

2. One or two triangular membrane properties may be defined on a single card.

Input Data Card PWEB WEB Property

Description: Defines the properties of 2 node WEB as referenced by the CWEB card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PWEB	PID	MID	T		PID	MID	T		
PWEB	235	2	0.5						

Field	Contents
PID	Property identification number (Integer >0)
MID	Material identification number (Integer >0)
T	Thickness of the shear web (Real >0.0)

Remarks:

1. All PWEB cards must have unique property identification numbers.
2. One or two WEB properties may be defined on a single card.

Input Data Card SPC Single-Point Constraint

Description: Defines sets of single-point constraints

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPC	SID	G	C		G	C			
SPC	2	32	231		5	1			

Field	Contents
SID	Identification number of single-point constraint set (Integer >0)
G	Grid point identification number (Integer >0)
C	Component number (any unique combination of the digits 1-3 (with no imbedded blanks) when point identification numbers are gridpoints).

Remarks:

1. Single-point constraint sets must be present in the input.
2. From one to twelve single-point constraints may be defined on a single card.
3. SPC degrees of freedom may be redundantly specified as permanent constraints on the GRID card.
4. The SID number should be the same on all SPC cards.

Input Data Card SPC1 Single-Point Constraint

Description: Defines sets of single-point constraints

Format and Example:

1	2	3	4	5	6	7	8	9	10
SPC1	SID	C	G1	G2	G3	G4	G5	G6	
SPC1	3	2	1	3	10	9	6	5	

Alternate form:

SPC1	SID	C	GID1	"THRU"	GID2
SPC1	313	12	6	THRU	32

Field	Contents
SID	identification number of single-point constraint set (Integer >0)
C	Component number (any unique combination of the digits 1-3 with no imbedded blanks) when point identification numbers are gridpoints.
G1, GID1	Grid point identification numbers (Integer >0)

Remarks:

1. Single-point constraint sets must be present in the input.
2. SPC degrees of freedom may be redundantly specified as permanent constraints on the GRID card.
3. All gridpoints referenced by GID1 thru GID2 must exist.
4. No NASTRAN "continuation" cards are allowed.

Input Data Card TITLE Title Card

Description: Information to be printed at the beginning of the computer listing.

Format and Example:

```
  1    2    3    4    5    6    7    8    9    10
TITLE
Plate Membrane Prob. No. 2
```

Input Data Card ENDDATA

Description: Defines the end of the data deck.

Format and Example:

```
  1    2    3    4    5    6    7    8    9    10
ENDDATA
ENDDATA
```

Remarks:

1. This card required even if no physical data cards exist in the deck.

2. ENDDATA must begin in columns 1 or 2.

3. Failure to include this card will result in an operating system termination caused by input end of file error.

3.3 Description of Output Data

The format of the output was designed with the engineering User in mind. Liberal use of labels was emphasized which serve to identify the output data and guide the User through the optimization solution procedure. Output print options are limited to those available on the OPTIM input card. This input permits printing of output for "bebug" purposes. The User should use this option when solutions are troublesome. A second option permits the printing of stresses and displacements at every iteration in the optimization process.

Typical output is given in Section 3.4. An echo print of the input data is always presented first. This is followed by output pertinent to the analysis being conducted such as shown in Section 3.4 for optimization solutions. Execution of static or dynamic analyses yields standard structural analysis data, ie. grid point displacements, element stresses and structural reactions. Vibration mode shapes are also given as required. These same data are given at the end of an optimization analysis in conjunction with the minimum weight value and associated design variable vector.

3.4 Illustrative Example

A seventeen bar cantilever truss structure is used to illustrate the input/output features of OPTFORCE II. Table 1 displays material properties, minimum sizes of design variables, allowable stress levels and loading condition. Figure 8 shows the geometric layout and finite element modeling of the truss. Input data, as written in NASTRAN format, is given in Figure 9. Each input card is described in detail in Section 3.2 of this report. The input data deck is very similar to that used in conducting any structural analysis using the NASTRAN computer code. In fact, the only new data in this application of the OPTFORCE II code are the OPLOADS, FORCE and OPTIM cards. The OPTIM card is particularly important since it controls analysis type, convergence criteria and output print options.

Problem output is given in Figure 10. The User's input data deck is reproduced for reference purposes. Selected output from the solution procedure is given next. Output statements and the use of Figure 4 should be sufficient to guide the User through the optimization procedure. When the final solution is obtained the minimum weight of the structure is given along with values of the design variables. In addition, the structures displacement, finite element stresses and reactions are given to complete the description of the anaylsis.

TABLE 1 MATERIAL PROPERTIES, CONSTRAINTS & LOADS - SEVENTEEN BAR TRUSS

(1) Material Properties

$E = 30.0 \times 10^6$ psi
ρ = .268 lbs. per cubic inch
ν = .30

(2) Minimum Size (Size Constraints)

Bar element area = .10 in^2

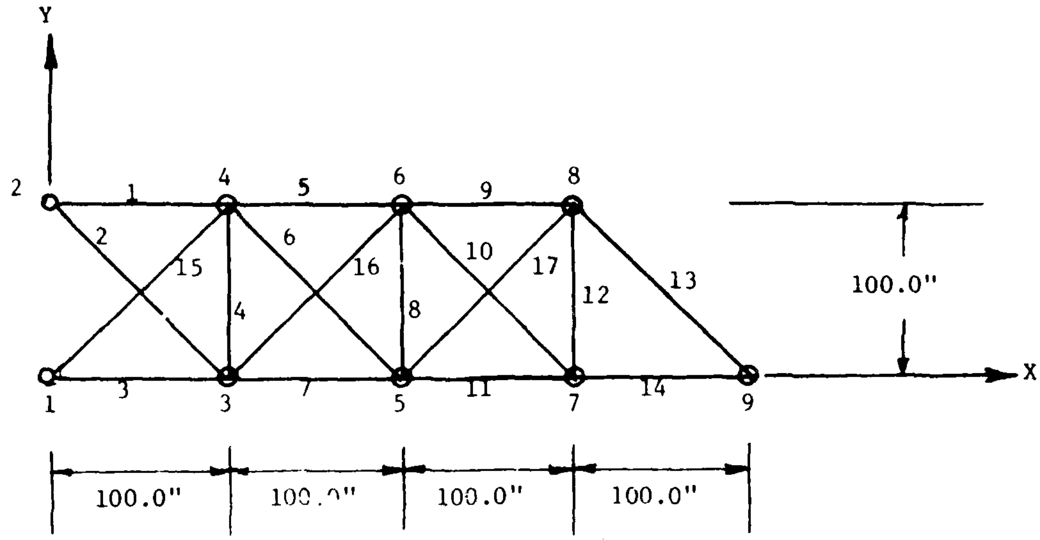
(3) Allowable Stress (Stress Constraints)

σ_l = -50000.0 psi, σ_u = 50000.0 psi
all elements except Nos. 2, 6, 10

σ_l = -125,000.0 psi, σ_u = 125,000.0 psi
elements Nos. 2, 6, 10

(4) Loading Condition (Single Loading Case)

Grid Pts.	3, 5, 7, 9
Direction	+Y
Value	-100,000.0 lbs.



Loads @	Grid Pts.	3, 5, 7, 9
	Direction	+Y
	Value	-100,000 lbs.

Figure 8 Seventeen Bar Truss

```
LIST
 10 TITLE
 11    SEVENTEEN BAR TRUSS-FOUR LOADS
 20 GRID    1                 0.0      0.0      0.0               123456
 30 GRID    2                 0.0      100.0    0.0               123456
 40 GRID    3                 100.0    0.0      0.0               3456
 50 GRID    4                 100.0    100.0    0.0               3456
 60 GRID    5                 200.0    0.0      0.0               3456
 70 GRID    6                 200.0    100.0    0.0               3456
 80 GRID    7                 300.0    0.0      0.0               3456
 90 GRID    8                 300.0    100.0    0.0               3456
100 GRID    9                 400.0    0.0      0.0               3456
110 OPLOAD$ 1                          1
150 OPTIM   YES     NO        YES                                          OPT
160 FORCE   1       3                  1.0+5    0.0      -1.0     0.0
170 FORCE   1       5                  1.0+5    0.0      -1.0     0.0
180 FORCE   1       7                  1.0+5    0.0      -1.0     0.0
190 FORCE   1       9                  1.0+5    0.0      -1.0     0.0
200 CONROD  1       2         4        1        0.10     0        0        0
210 CONROD  2       2         3        2        0.10     0        0        0
220 CONROD  3       1         3        1        0.10     0        0        0
230 CONROD  4       3         4        1        0.10     0        0        0
240 CONROD  5       4         6        1        0.10     0        0        0
250 CONROD  6       4         5        3        0.10     0        0        0
260 CONROD  7       3         5        1        0.10     0        0        0
270 CONROD  8       5         6        1        0.10     0        0        0
280 CONROD  9       6         8        1        0.10     0        0        0
290 CONROD  10      6         7        4        0.10     0        0        0
300 CONROD  11      5         7        1        0.10     0        0        0
310 CONROD  12      7         8        1        0.10     0        0        0
320 CONROD  13      8         9        1        0.10     0        0        0
330 CONROD  14      7         9        1        0.10     0        0        0
340 CONROD  15      1         4        1        0.10     0        0        0
350 CONROD  16      3         6        1        0.10     0        0        0
360 CONROD  17      5         8        1        0.10     0        0        0
370 MAT1    1       30.+6              0.3      0.268                               +MATA
380 +MATA   -5.0+4  +5.0+4
390 MAT1    2       30.+6              0.3      0.268                               +MATB
400 +MATB   -12.5+4 +12.5+4
410 MAT1    3       30.+6              0.3      0.268                               +MATC
420 +MATC   -12.5+4 +12.5+4
430 MAT1    4       30.+6              0.3      0.268                               +MATD
440 +MATD   -12.5+4 +12.5+4
450 ENDDATA
```

Figure 9 OPTFORCE II Input Data - Seventeen Bar Truss

```
CARD COUNT OF INPUT DATA
  CARD TYPE    NUMBER
   BEGIN BU       1
   TITLE          1
   BUCK           0
   CONROD        17
   CTUBEAM        0
   CQDMEM1        0
   BUCK1          0
   CQUAM8         0
   CRDM1D         0
   CROD           0
   CSHEAR         0
   CTRIM6         0
   CTRMEM         0
   GCON           0
   CWEB           0
   FORCE          4
   GRID           9
   ICON           0
   LINKS          0
   MAT1           4
   MOMENT         0
   OPLOADS        1
   GPCEQN         0
   OPDVIR         0
   OPTIM          1
   PQDMEM1        0
   PRDM1D         0
   PROD           0
   PSHEAR         0
   PTRMEM         0
   PTUBEAM        0
   SPC            0
   SPC1           0
   SPC1           0
   PWEB           0
   PTRIM6         0
   PQUAM8         0
   ENDDATA        1
   PTUBEAM        0
   MAT2           0
BEGIN BU      LK
TITLE
  SEVENTEEN BAR TRUSS-FOUR LOADS
GRID          1            0.0       0.0       0.0              123456
GRID          2            0.0   100.000       0.0              123456
GRID          3        100.000       0.0       0.0                3456
GRID          4        100.000   100.000       0.0                3456
GRID          5        200.000       0.0       0.0                3456
GRID          6        200.000   100.000       0.0                3456
GRID          7        300.000       0.0       0.0                3456
GRID          8        300.000   100.000       0.0                3456
GRID          9        400.000       0.0       0.0                3456
OPLOADS       1      0               1         0          0          0
OPTIM       YES     NO     YES       0         0 0.0      0.0             OPT
FORCE         1      3      0.100E+06    0.0    -1.0000   0.0
FORCE         1      5      0.100E+06    0.0    -1.0000   0.0
FORCE         1      7      0.100E+06    0.0    -1.0000   0.0
FORCE         1      9      0.100E+06    0.0    -1.0000   0.0
CONROD        1      2      4        1    0.100       0          0          0
CONROD        2      2      3        2    0.100       0          0          0
```

Figure 10 OPTFORCE II Output - Seventeen Bar Truss

```
CONROD       3       1       3       1   0.100       0       0       0
CONROD       4       3       4       1   0.100       0       0       0
CONROD       5       4       6       1   0.100       0       0       0
CONROD       6       4       5       3   0.100       0       0       0
CONROD       7       3       5       1   0.100       0       0       0
CONROD       8       5       6       1   0.100       0       0       0
CONROD       9       6       8       1   0.100       0       0       0
CONROD      10       6       7       1   0.100       0       0       0
CONROD      11       5       7       1   0.100       0       0       0
CONROD      12       7       8       1   0.100       0       0       0
CONROD      13       8       9       1   0.100       0       0       0
CONROD      14       7       9       1   0.100       0       0       0
CONROD      15       1       4       1   0.100       0       0       0
CONROD      16       3       6       1   0.100       0       0       0
CONROD      17       5       8       1   0.100       0       0       0
MAT1         1  30.+6                0.3   0.260                   +MATA
    +MATA  -5.0+4  +5.0+4
MAT1         2  30.+6                0.3   0.268                   +MATB
    +MATB -12.5+4 +12.5+4
MAT1         3  30.+6                0.3   0.268                   +MATC
    +MATC -12.5+4 +12.5+4
MAT1         4  30.+6                0.3   0.268                   +MATD
    +MATD -12.5+4 +12.5+4
ENDDATA
```

Figure 10 (cont'd.)

```
                                         INPUT DESIGN VARIABLES
          DESIGN VARIABLES
ELEMENTS
  1- 10  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00
 11- 17  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00  0.1000E+00

                   TOTAL STRUCTURE WEIGHT =  0.5333055E+02


ENTER STRESS RATIO METHOD


                                    INITIAL GUESS - STRESS RATIO METHOD

          DESIGN VARIABLES
ELEMENTS
  1- 10  0.1591E+02  0.2316E+01  0.1605E+02  0.7176E+00  0.3811E+01  0.1804E+01  0.[illegible]85E+01  0.7273E+00  0.3916E+01  0.1179E+01
 11- 17  0.4084E+01  0.1000E+00  0.2828E+01  0.2000E+01  0.5524E+01  0.3976E+01  0.2710E+01

                   TOTAL STRUCTURE WEIGHT =  0.2420172E+04


ENTER STATIC ANALYSIS
ENTER ITER/LINEAR PROGRAMING SUBROUTINE
```

Figure 10 (cont'd.)

```
                         LINEAR PROGRAMING PHASE - ITERATION NO.1

       SOLUTION IS FEASIBLE

 ENTER PARTIAL N-R ROUTINE
            DESIGN VARIABLES
 ELEMENTS
   1- 10  0.1591E+02  0.2316E+01  0.1609E+02  0.1000E+00  0.8811E+01  0.1804E+01  0.9189E+01  0.1000E+00  0.2516E+01  0.1179E+01
  11- 17  0.4084E+01  0.1000E+00  0.2828E+01  0.2000E+01  0.5524E+01  0.3976E+01  0.2710E+01

                    TOTAL STRUCTURE WEIGHT =  0.2386811E+04

       ITERATION NO.  1
            DESIGN VARIABLES
 ELEMENTS
   1- 10  0.1610E+02  0.3175E+01  0.1550E+02  0.1000E+00  0.7935E+01  0.2484E+01  0.9946E+01  0.1000E+00  0.4054E+01  0.1319E+01
  11- 17  0.3854E+01  0.1000E+00  0.2628E+01  0.2000E+01  0.5785E+01  0.2479E+01  0.2953E+01

                    TOTAL STRUCTURE WEIGHT =  0.2409178E+04

       ITERATION NO.  2
            DESIGN VARIABLES
 ELEMENTS
   1- 10  0.1610E+02  0.3608E+01  0.1550E+02  0.1000E+00  0.7999E+01  0.2786E+01  0.1000E+02  0.1000E+00  0.4100E+01  0.1343E+01
  11- 17  0.3900E+01  0.1000E+00  0.2628E+01  0.2000E+01  0.5798E+01  0.2785E+01  0.2970E+01

                    TOTAL STRUCTURE WEIGHT =  0.2454361E+04

       ITERATION NO.  3
            DESIGN VARIABLES
 ELEMENTS
   1- 10  0.1610E+02  0.3676E+01  0.1550E+02  0.1000E+00  0.8000E+01  0.2828E+01  0.1000E+02  0.1000E+00  0.4100E+01  0.1344E+01
  11- 17  0.3900E+01  0.1000E+00  0.2828E+01  0.2000E+01  0.5798E+01  0.2828E+01  0.2970E+01

                    TOTAL STRUCTURE WEIGHT =  0.2460140E+04

       ITERATION NO.  4
            DESIGN VARIABLES
 ELEMENTS
   1- 10  0.1610E+02  0.3677E+01  0.1550E+02  0.1000E+00  0.8000E+01  0.2828E+01  0.1000E+02  0.1000E+00  0.4100E+01  0.1344E+01
  11- 17  0.3900E+01  0.1000E+00  0.2828E+01  0.2000E+01  0.5798E+01  0.2828E+01  0.2970E+01

                    TOTAL STRUCTURE WEIGHT =  0.2460236E+04

       DESIGN CONVERGED IN PARTIAL N-R ROUTINE
```

Figure 10 (cont'd.)

```
                              LAGRANGE MULTIPLIER CHECK

     LAMBDA - DESIGN VARIABLES(1- 17)
0.0        0.0        0.0        0.4466E+01 0.0        0.0        0.0        0.3573E+01 0.0        0.0
0.0        0.4467E+01 0.0        0.0        0.0        0.0        0.0

     LAMBDA - STRESS(1- 17) - LOAD CONDITION  1
0.4779E+03 0.0        0.3797E+03 0.7147E+02 0.2412E+03 0.0        0.2412E+03 0.3064E+02 0.1226E+03 0.0
0.4179E+02 0.1044E+02 0.1072E+03 0.5360E+02 0.3127E+03 0.1608E+03 0.1380E+03

     LAMBDA - REDUNDANTS(1-  3) - LOAD CONDITION  1
0.3942E+03 0.2274E+03 0.1080E+03

     DESIGN, STRESS AND DISPLACEMENT LAMBDAS ARE ALL GREATER THAN ZERO
       EXIT WITH LAST DESIGN
```

Figure 10 (cont'd.)

FINAL SCLUTION

DESIGN VARIABLES

ELEMENTS

1- 10	0.1610E+02	0.3677E+01	0.1550E+02	C.1000E+00	0.8000E+01	0.2828E+01	0.1000E+02	0.1000E+00	C.4100E+01	0.1344E+01
11- 17	0.3900E+01	0.1000E+C0	0.2828E+01	C.2000E+01	0.5798E+01	0.2828E+C1	0.2970E+01			

TCTAL STRUCTURE WEIGHT = 0.2460236E+04

ENTER STATIC ANALYSIS

DISPLACEMENTS---FOR LOAD CONDITICN NO. , 1

NODE NO	X	Y	Z
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	-1.6666669E-01	-6.6666424E-01	0.0
4	1.6666663E-01	-5.0000572E-01	0.0
5	-3.3333331E-01	-1.6666670E+00	0.0
6	3.3333308E-01	-1.5000029E+00	0.0
7	-5.0000000E-01	-2.9999933E+00	0.0
8	4.9999976E-01	-2.8333149E+00	0.0
9	-6.6666675E-01	-4.3333206E+00	0.0

ELEMENT STRESSES---LOAD CONDITICN NC. 1

ELEM NO.	ELEM TYPE	SX	SY	TXY	YI/A(I)
1	1	5.0000012E+04			5.0000012E+04
2	1	7.4999687E+04			7.4999687E+04
3	1	-5.0000059E+C4			5.0000059E+04
4	1	4.9998109E+C4			4.9998109E+04
5	1	5.0000004E+04			5.0000004E+04
6	1	9.9999937E+04			9.9999937E+04
7	1	-5.0000016E+04			5.0000016E+04
8	1	4.9999359E+04			4.9999359E+04
9	1	5.0000016E+C4			5.0000016E+04
10	1	9.9999312E+04			9.9999312E+04
11	1	-5.0000031E+C4			5.0000031E+04
12	1	4.9999984E+04			4.9999984E+04
13	1	5.0000047E+04			5.0000047E+04
14	1	-5.0000047E+04			5.0000047E+04
15	1	-4.9999996E+C4			4.9999996E+04
16	1	-4.9999980E+C4			4.9999980E+04
17	1	-5.0000016E+04			5.0000016E+04

REACTIONS----FOR LUAD CCNDITICN NO. , 1

NODE NO	X	Y	Z
1	9.9999950E+05	2.0499987E+05	0.0
2	-9.9999950E+05	1.9500012E+05	0.0

Figure 10 (cont'd.)

3	0.0	0.0	0.0
4	0.0	0.0	0.0
5	0.0	0.0	0.0
6	0.0	0.0	0.0
7	0.0	0.0	0.0
8	0.0	0.0	0.0
9	0.0	0.0	0.0

END OF FILE ON READ IN SUBR READI. END OF PROBLEM

Figure 10 (concluded)

(This page blank)

APPENDIX A

PROGRAMMER'S MANUAL FOR OPTFORCE II

Dennis Witkop
Steve Skalski

TABLE OF CONTENTS

LIST OF FIGURES

A.1 *INTRODUCTION*

The programming aspects of OPTFORCE II are described in this Appendix. The information presented here is geared to the Programmer. It is sufficient to fully describe the general program logic and the required peripheral storage. All element generated data is stored externally to reduce core storage. A separate section is devoted to the description of these files so that I/O time may be optimized through efficient buffer description. Individual subroutine write-ups are presented along with the complete Fortran source listing.

A short description of each routine is included to aid in obtaining an overall familiarity with the program's components.

Finally, a discussion is provided for the concept of dynamic storage which allows the program to execute in a variable storage environment.

A.2 GENERAL PROGRAM LOGIC

The general organization of the OPTFORCE II program is illustrated in Figures A.1 thru A.5. The computer program consists of a control program which includes four principal phases:

1. Data Interpretation Phase
2. Initialization Phase
3. Structure Cutter Phase
4. Calculation Phase

A.2.1 Control Program (Figure A.1)

The "MAIN" routine controls the program flow. The number of work storages in the entire program is determined by the variable NWORK. This parameter can be adjusted to fit the storage requirements of the problem. LINK1 consists of phase 1 and 2 - Data Interpretation Phases and Initialization Phase. LINK2 consists of the phases 3 and 4 - Structure Cutter phase and Calculation Phase.

A.2.2 Data Interpretation Phase (Figure A.2)

The principal purpose of this phase is to read the NASTRAN form input data and prepare the information for the next three phases. Subroutines SORT, ZZ, and OPINPT perform this function. An echo print of this input is also provided here. The SORT and ZZ routines sort and count the NASTRAN data records and then form the dynamic storage constants. Each NASTRAN label card is processed and stored in core by "ZZ". An input file data unit NSS1 is then written. This data file is used by OPINPT which writes on unit NTAPE all the data needed for initialization.

A.2.3 Initializaton Phase (Figure A.3)

This phase produces a variety of information for the Structure Cutter and Calculation Phase. Upon being activated by the OPTIM2 routine, the "NEWS" routine determines the dynamic storage allocation for the Initialization Phase. The "AONE" routine is then activated and performs the following tasks:

a. A load matrix is generated in reduced form and written on file NSS1.

b. Flexibility and FORCE matrices are defined by the called element routines (ELEM1-ELEM4) and are written on file NSS1. The number of forces in the element and the weight of the element with a unit design variable are also written on file NSS1.

A.2.4 Structure Cutter Phase (Figure A.4)

This phase produces four data sets (I11, I12, I13 and I15) for the Calculation Phase. Routine TAPE11 defines the structure cutter matrix from

information on unit NSS1. As this process continues, unit I11 is generated. Upon calling routine AA, the matrix is solved. Finally, routine S241 controls the generation of the basic structural matrices $\bar{\Phi}$, $\bar{\Psi}$ and $\bar{\Omega}$ for each element in the structure. These matrices and information about the mass matrix is written to unit I13, the b_1 matrix is written to unit I12 and the D_1 matrix is written to unit I15.

A.2.5 Calculation Phase (Figure A.5)

The purpose of phase 4 is to perform all computations required for static analysis, dynamic analysis or structural optimization. Input files required by this phase are FILE I12, I13 and I15 while formatted output is written to FILE I01 and I02 (line printer). Phase 4 (OPTFR) is called by routine "LINK2" which reads all element and control data from FILE I11 and allocates dynamic storage for phase 4. Subroutine "OPTFR" is the control and initialization routine for the phase 4 routines.

Selection of static or dynamics analysis by the user results in output of the design variables followed by either a static analysis (BASIC) or dynamic analysis (BASICD) and output of the computed results. Control is then returned to routine LINK2.

User selection of structural optimization follows the flow path as depicted in Figure A.5 (optimization) and is described as follows:

a. Initial Guess - User input design variables may be used as the initial guess or used as input to the FSD routine to compute a new design using the stress ratio method. The initial guess is printed and a basic analysis is performed by subroutine "BASIC".

b. Compute Coefficients - Subroutines S432, S435, S4310, S4323, S4325 and S4310A are used to compute all $\partial g/\partial A$, $\partial g/\partial X$ values and assemble them in a coefficient matrix which is input to the linear programming routine "ITER".

c. Linear Programming - The function of subroutine ITER is to determine if a linear solution is feasible for minimization of the objective function which is subjected to the equality constraints $\partial L/\partial A=0$ and $\partial L/\partial X=0$. In addition the Lagrange multipliers (μ_A, μ_S, μ_D and μ_W) are constrained to be >0. If a feasible solution does exist the active constraints (μ>0.0) are used in subroutine S451 to solve for a new design. The linear equations $(A_{e+1}-A_e)\frac{\partial g}{\partial A}+g=0$ and $(X_{e+1}-X_e)\frac{\partial g}{\partial x}+gx=0$ are solved simultaneously for new design A_{e+1} and redundant X_{e+1} using routine SIMQ. Subroutine S451 is re-iterated until convergence or a maximum number of iterations is reached. If the design converges and all constraints are satisfied, linear programming is exited, otherwise, a maximum of 2 additional linear programming iterations are made. Whenever 3 linear programming iterations have been completed without convergence or "ITER" indicates an infeasible design a full stressed design is computed and output.

After the statically constrained design has converged or a fully stressed design is generated any requested dynamic constraints are included and a dynamic analysis is performed using subroutine "BASICD". Linear

programming is then re-entered to obtain a new design with dynamic constraints.

d. Compute Lagrange Multipliers - The design obtained from linear programming is used to compute the Lagrange multipliers for all active constraints using subroutine "EQSOL". If all of the Lagrange multipliers are ≥ 0.0 and constraint equations are satisfied, optimization is complete and the final soltuion is printed.

e. Newton-Raphson Procedure - In the cases where there are negative Lagrange multipliers or constraint violations, a full Newton-Raphson is performed by Executing subroutine S461. Subroutine S461 is an iterative routine which takes derivatives of $\partial L/\partial A$, $\partial L/\partial X$ and g to obtain coefficients for a set of linear equations. These equations are solved simultaneously for all ΔA, ΔX and $\Delta\mu$ by subroutine "GELS". The iterative process is terminated when convergence of all variables is achieved or a maximum number of iterations has been reached.

f. Output of Final Design - A print of the final design and engineering parameters is accomplished by routines PRIN1, BASIC and if dynamic constraints are included BASICD is executed.

Subroutine OPFTR then returns to LINK2 for program termination.

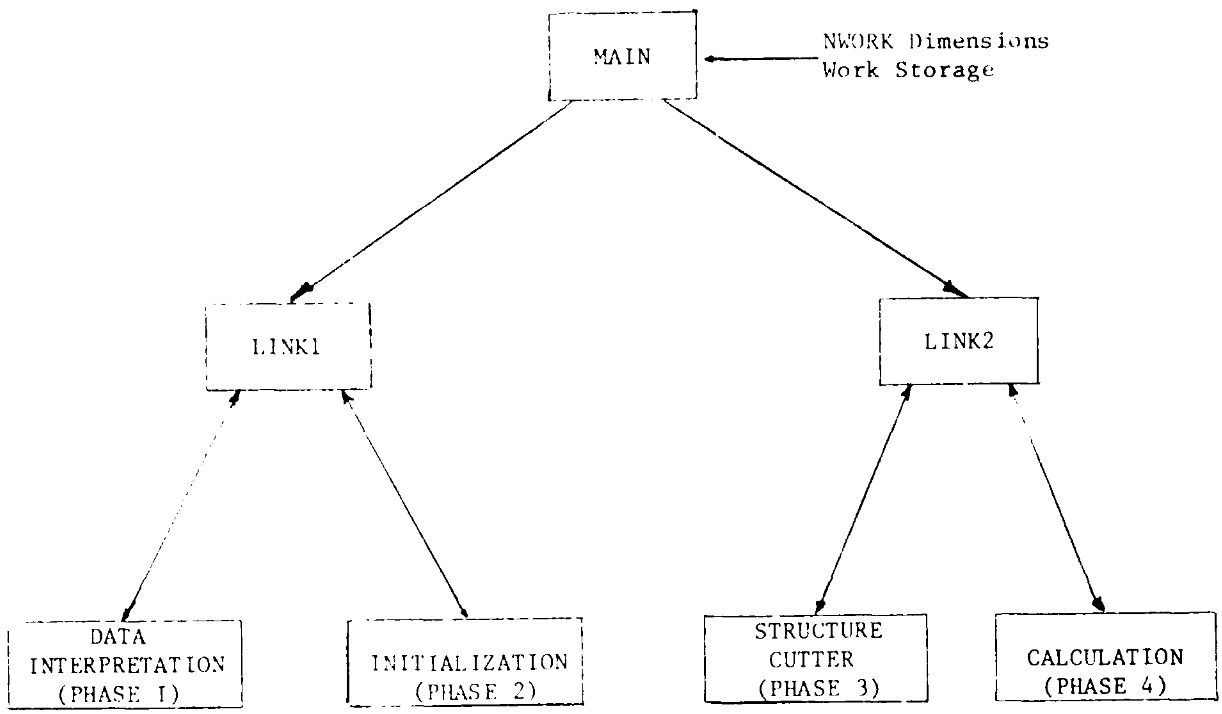


FIGURE A.1 CONTROL PROGRAM

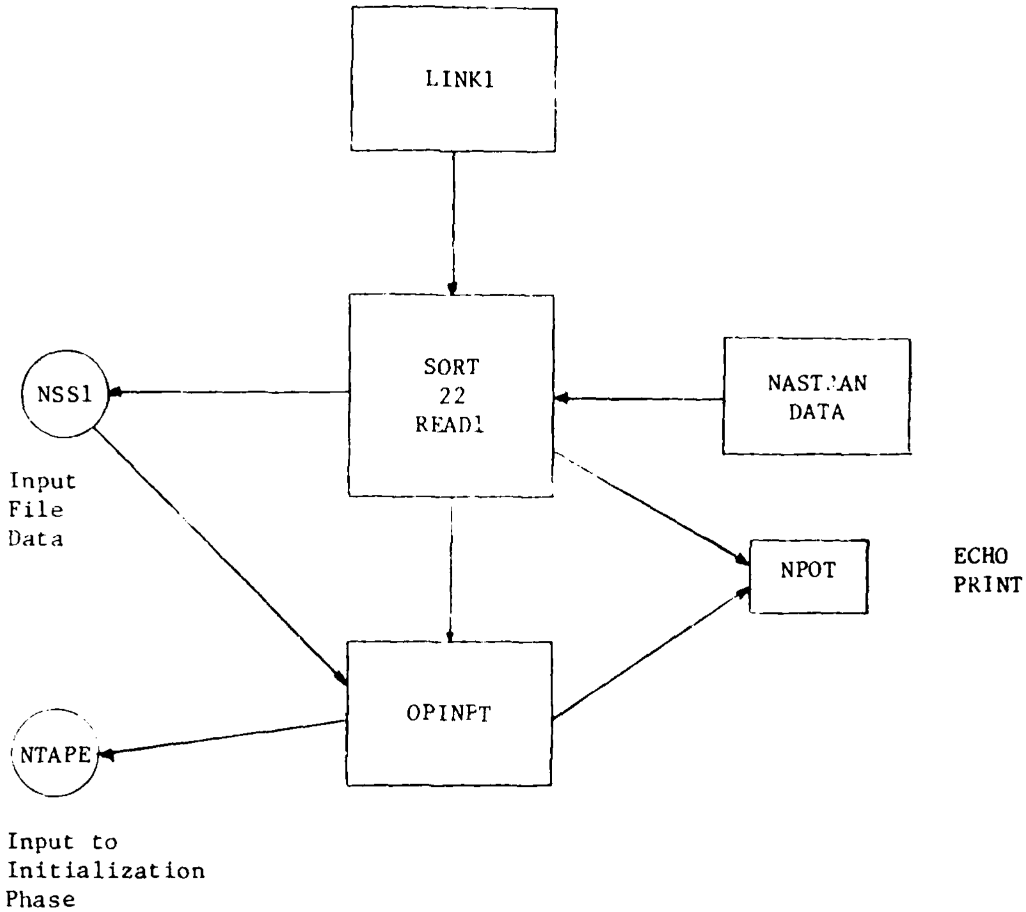


FIGURE A.2 DATA INTERPRETATION PHASE

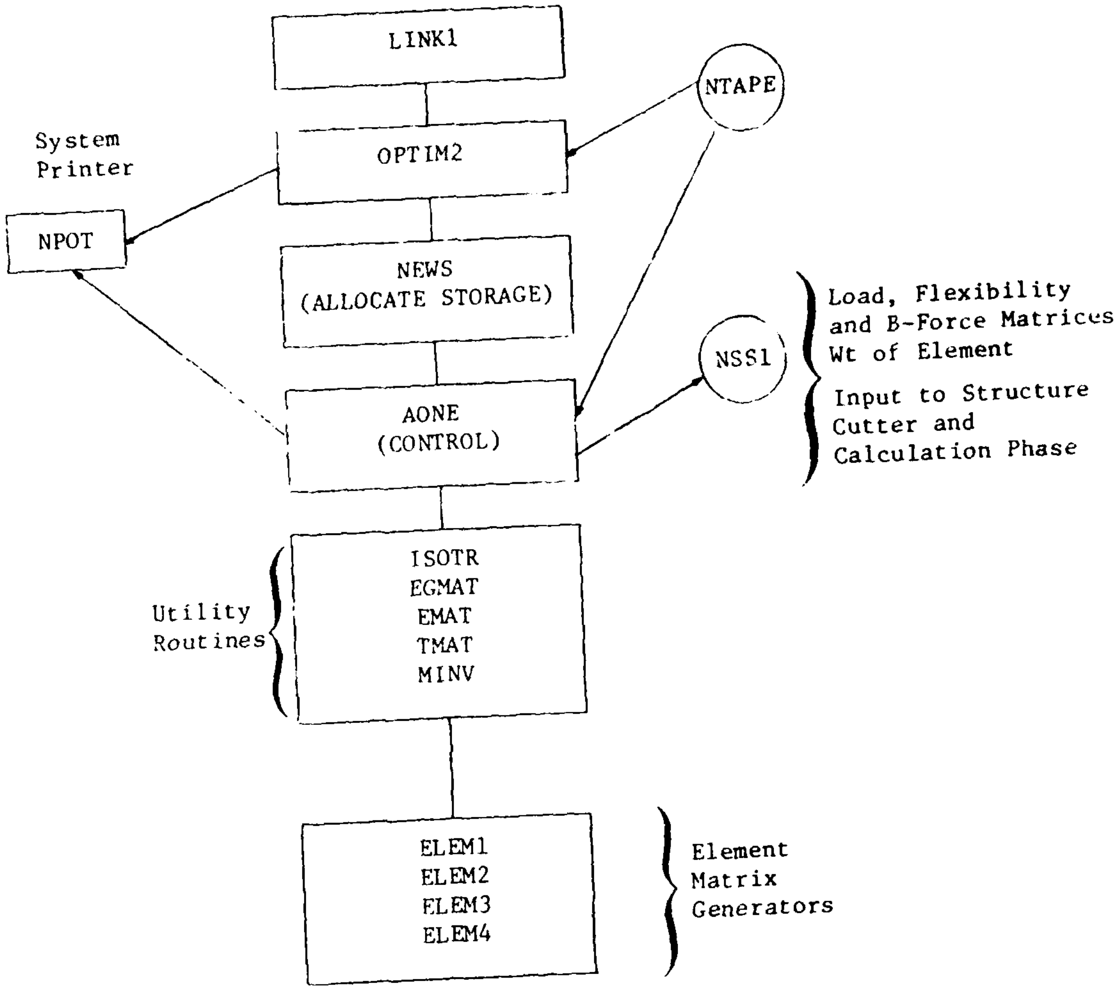


FIGURE A.3 INITIALIZATION PHASE

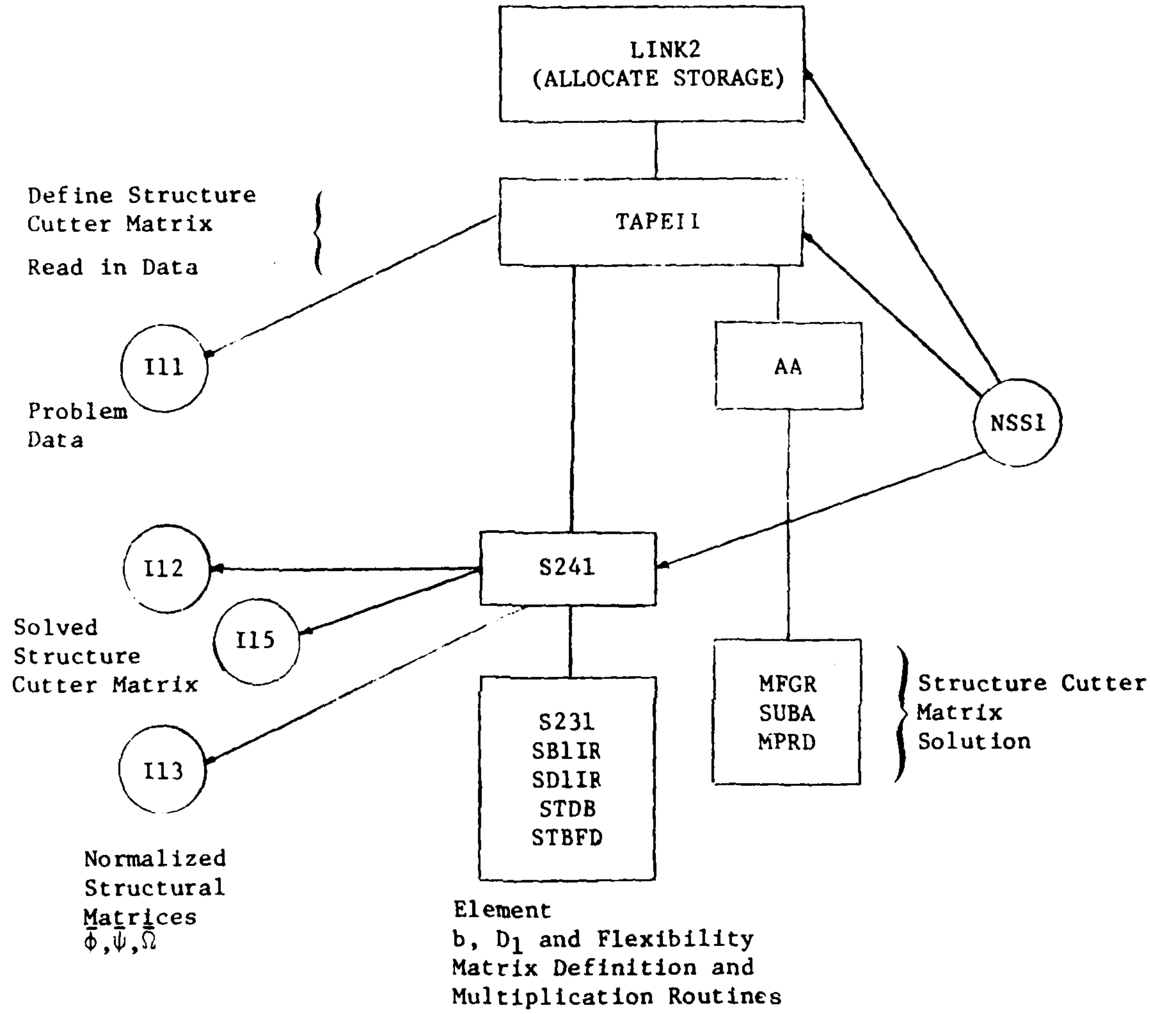


FIGURE A.4 STRUCTURE CUTTER PHASE

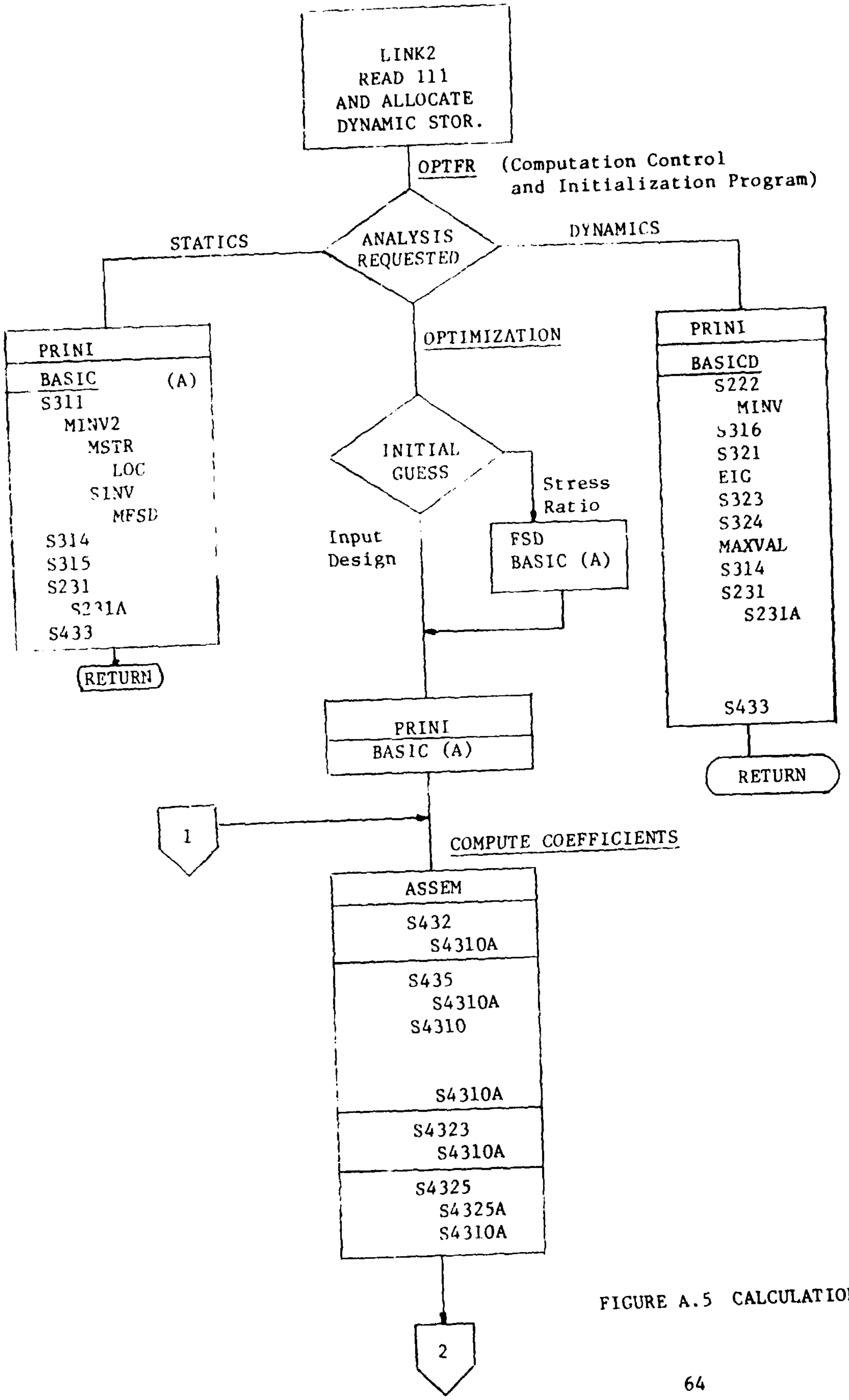


FIGURE A.5 CALCULATION PHASE

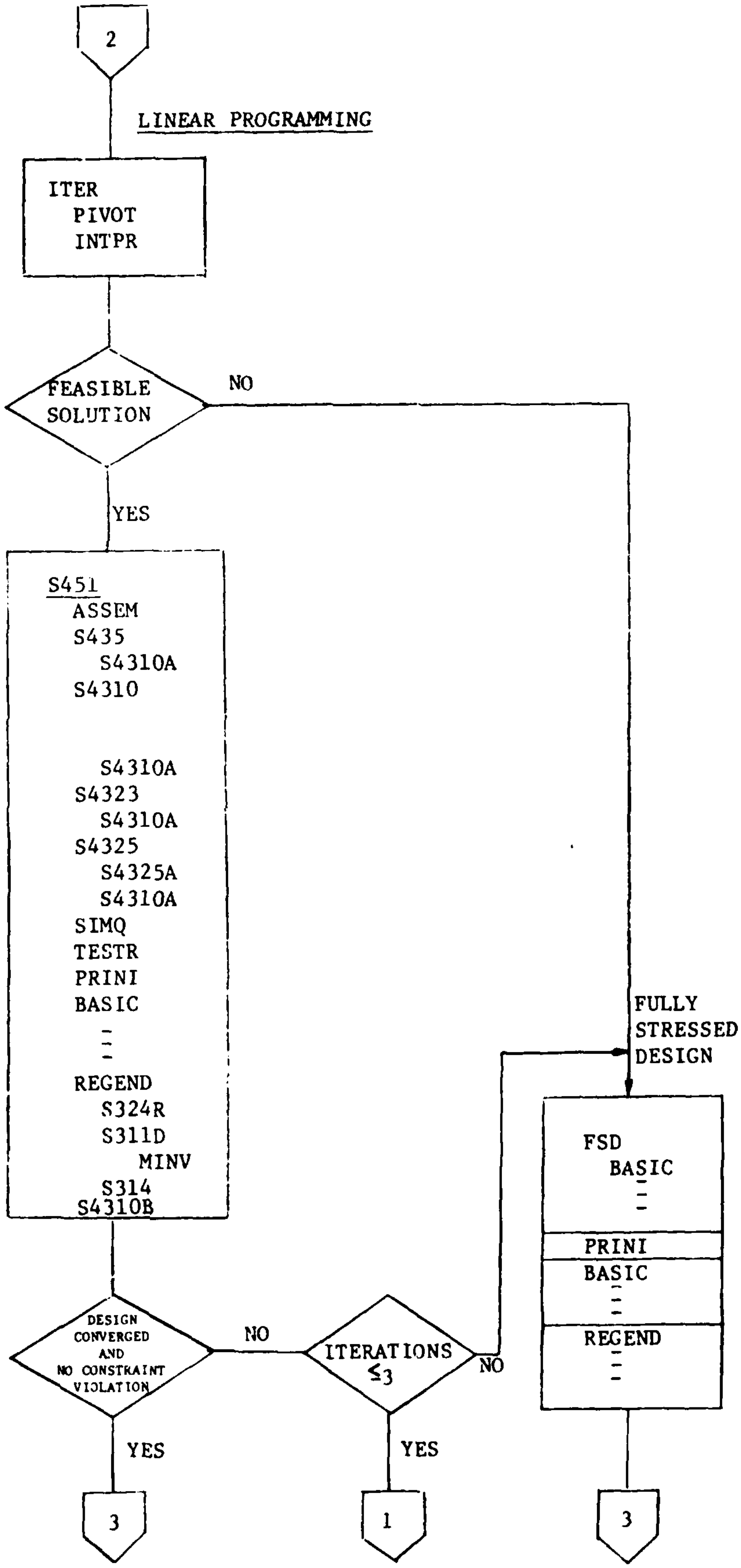


FIGURE A.5 CALCULATION PHASE

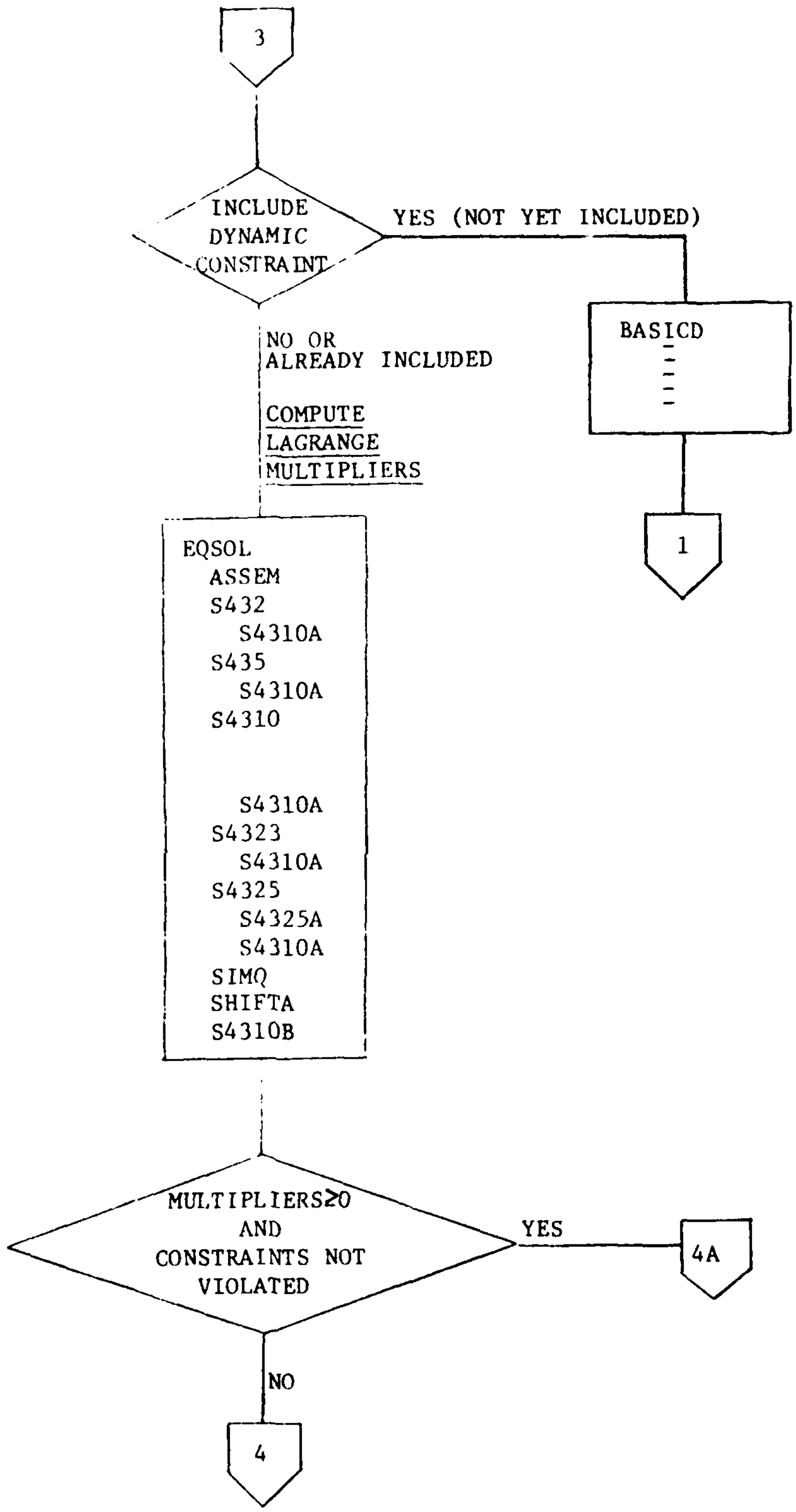


FIGURE A.5 CALCUATION PHASE

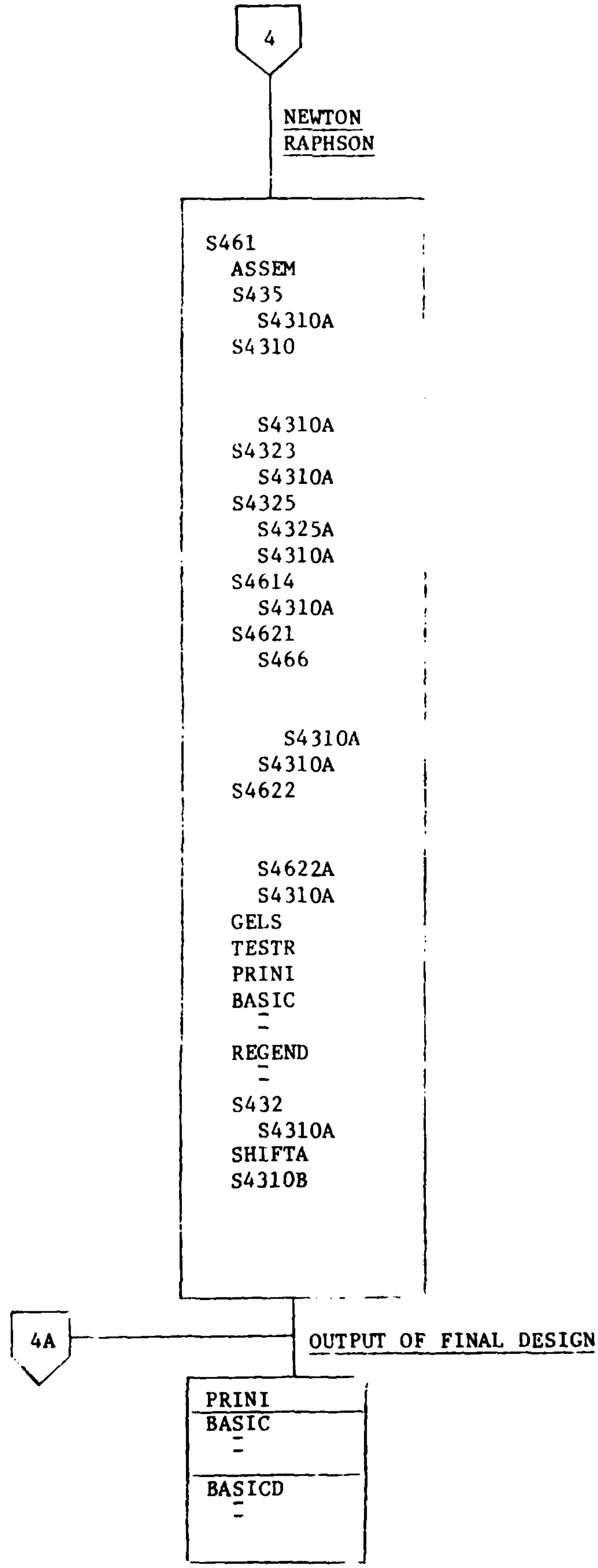


FIGURE A.5 CALCULATION PHASE

A.3 EXTERNAL DATA SET STRUCTURE

This program uses seven data sets during execution. The delivery version of the OPTFORCE II program comes with the following variable names and real unit designations.

Unit Name	Unit Id	Usage
NTAPE	13 (BINARY)	Output from "MAIN" and "OPINPT" routines and input to "AONE" routine.
NSS1 NSS2 NSS3 NSS4	1	Contains load matrix and element flexibility matrices with assembly information. Contains element B-Force matrices. ITOT (no. of elements) pairs of information from element matrix generation routines used to initialize the calculation phase.
I11	11 (BINARY)	Output from "TAPE11" routine and used in calculation phase contains general description of structure
I12	12 (BINARY)	Output from "S241" routine contains b_1 matrix resulting from structure cutter.
I13	13 (BINARY)	Output from "S241" routine contains all bar (element) matrices of PHI, PSI and Omega.
I14	14 (BINARY)	Output from "S4310A" routine. contains matrix element row, column, data, transpose/symmetry control and summing control - scratch file.
I15	15 (BINARY)	Output from "S241" routine contains D_1 matrix resulting from structure cutter.
L4	9 (FORMATTED)	Output from "SORT" and input to "ZZ". Contains identification record and NASTRAN input card image.
L7	7 (FORMATTED)	Output from "ZZ" and input to "OPTIM2". Contains card images in OPTIM MAGIC format.
NPIT/J5/L5	5	Standard card input (80 column card)

NPOT/JE6/J6/ L6/IO2	6	Standard Line Printer (132 characters/line).
IO1	10	Debug print (132 character/record).

NOTE: Since units 5 and 6 are standard input and output, they will not be considered in this discussion.

NTAPE CONTENTS

Note: This unit is called I5 in routine AONE.

NTCDS Records (Title Cards)	
One Record (Coordinates)	(X(I),Y(I),Z(I),I=1,N2
One Record Per Element (ITOT Records) (Element Data)	IE,IELT(I),IBUCKL,N5(I),N6(I),N7(I),N8(I),N11(I),N13(I) N15(I),N17(I),(EM(II,I),II=1,11),JMAT(I),ANGLE(I)
One Record (DOF Bounded)	(KL(I),I=1,NBOU)
NN2L Records (Load Components)	IR,IC,C1,C2
IF NDL.NE.0 Individual Constraint (DOF-ICON)	(NBDF(I),I=1,NDL
One Record If NDL.NE.0 (Displacement Limits)	(DISPU(I),DISPL(I),I=1,NDL
If IRST .EQ1 or 2 or 5 One Record (Input Element Design Paramets)	(ALL(I),I=1,I TO T)

A.3.1 NSS1 CONTENTS

1st Record (Problem Control Data)	NRDF,NLOAD,ITOT,NALD,NDL,NSG,NAA,NBOV,NDOF,N2,NELI,IRST, NMODES
2nd Record (Stress Limits Min Size Element Type)	(D,SIGU(I),ALL(I),I=1,ITOT),IBS,(D,I=1,IBS) ,((D,I=1,11),J=1,ITOT),(IELT(I),I=1,ITOT)
Dummy Read If NDL NE 0	(DISPU(I),DISL(I),I=1,NDL
(Displacement Constraint DOF's)	(N22I(I)=I-1,NDL
If NDL NE 0 (Displacement Limits)	(DISPU(I),DISL(I),I=1,NDL
Program Flow Controls	IREST,IPRI,IPRS,ISTRM,IOO,MAX50,NLINK1,NICON NGRUPS,NODES4,NOSYMM,NANTI,NNONS,COOO5,WISTAR,IRST,NMODES
Load Matrix	J1,(DELTA(I),I=1,J1),NBOU,(KL(I),I=1,NBOU)
1 Record Per Element (Element Matrices and Data)	KS,KEL,AREA(NEL),(LI3(I),I=1,KEL ,((B(I,J),I=1,KEL),J=1,KS), D,D,D,D,((F(I,J),I=1,KS),,J=1,KS),D,NNE,((D,I=1,NNE),J=1,NNE) (D,I=1,KEL),(D,I=1,KEL),NNNO,D,D,D,D,D,D,D,D,D, WT(NEL),XC(NEL),YC(NEL)
If IRST NE 0 or NE 3 or NE 4 or NE 6	ALL2(I),I=1,ITOT
Input Starting Design Parameters (OPDVIR Card)	

Written by Routine AONE
Read by Routines TAPE11 and S241

A.3.2 I11 CONTENTS

1st Record (Problem Constants)	M,J1,NRED,NBOU,NDOF,N2,NELI,NDTNX,NDL,IRST,NMODES
2nd Record (Element Data Size and Stress Limits)	(IELT(I),ALL(I),WT(I),AREA(I),XC(I),YC(I),ALL2(I),SIGU(I),I=1,ITOT)
3rd Record (Load Matrix and Assembly Data)	(DELTA(I),I=1,J1),(DL(J),J=1,NBOU)
4th Record (Displacement Constraints and DOF Affected)	(DISPU(I),DISPL(I),N221(I),I=1,NDL

A.3.3 I12 CONTENTS

One Record Per Element (BI_I Matrix)	(BII(J), J=1, (5*NX))
One Record (BI_R Matrix)	(BIR(J),J=1, (NBOU*NX))

Written by S241 routine

A.3.4 I13 CONTENTS

One Record Per Element (Normalized Structural Bar Matrices and Mass Matrix Data)	((PHIB(I,J),I=1,NX),J=1,NX), ((PSIB(I,J),I=1,NX),J=1,NDN), ((OMEGAB(I,J),I=1,NDN),J=1,NDN) (Z(I),I=1,NDN)

Written by S241

A.3.5 I14 CONTENTS

One Record Per Matrix Element (Matrix - Row Number, Column Number, Transpose/ Symmetry Control, Element Data, Summary Control)	IB1,IB2,NT,ANS,NS

Scratch file written by S431CA routine

A.3.6 I15 CONTENTS

One Record Per Element ($D1_I$ Matrix)	(D1I(J),J=1,(5*NDN))
One Record ($D1_R$ Matrix)	(DIR(J),J=1,(NBOU*NDN))

Written by S241 routine

A.3.7 L4 CONTENTS (80 Character card images)

These records can be in any order on the file. It represents data in "pairs" of records. Each pair represents one specific type of data items determined by the value "NKIND" and "NCOUNT" which are written on the first record of the pair. R

Record one of the pair: NKIND, NCOUNT

Record two is determined by NKIND as shown in the following table:

NKIND	Second Record	
1	BULK	data
2	TITLE	"
4	CONROD	"
6	CQDMEM1	"
8		"
12	CROD	"
13	CSHEAR	"
16	CTRMEM	"
19	END*	"
20	ENDDATA	"
21	FORCE	"
23	GRID	"
24	ICON	"
27	MAT1	"
28	OPLOAD	"
30	OPDVIR	"
31	OPTIM	"
33	PQDMEM1	"
39	PROD	"
40	PSHEAR	"
43	PTRMEM	"
46	SPC	"
47	SPC1	"
48	SPC1 THRU	"
49	CWEB	"
50	PWEB	"
51	BEGIN BULK	"

A.4 Subroutine Write-Ups

Each subprogram of OPTFORCE II is described in this section. Included with each description is a statement declaring the size of the subprogram. This number is intended as a guide only, as it reflects the storage requirement on an IBM/370/3031, FORTRAN compiler.

In this manual the subroutine write-ups are presented in alphabetical order, separately for LINK1 routines and LINK2 routines.

A.4.1 BRIEF ROUTINE DESCRIPTIONS, LINK1

Routine Name	Purpose
ADJUST	Adjust each NASTRAN form field so that it is either left or right adjusted.
AONE	Control initialization
EGMAT	Generates EG matrix for triangular and quad plates
EMAT	Generate elasticity matrix for either orthotropic or isotropic properties, membrane triangle, quad
ELEM1	Element Matrix Generators: Axial
ELEM2	Element Matrix Generators: Shear web
ELEM3	Element Matrix Generators: Triangle
ELEM4	Element Matrix Generators: Quad
EXIT	Provide for final stop in program
FOMO	Process Force Cards
INSPC	Stores data obtained from input card into working storage
ISOTR	Given any two of G, E and μ computes third quantity
LINK1	Control program flow for PHASE1 and PHASE2
MINV	Matrix Inverse
NEWS	Allocate storage for initialization
OPINPT	Interpret report from input and place results on NTAPE
OPTIM2	Interprets input and initializes data - NEWS, OPINPT, EXIT
READI	Reads and modifies input data
SORT	Sort and count data based on LABEL information
SPCSUB	Process SPC (single point constraint) cards
TMAT	Generate transformation matrix for triangular plate orthotropic angle
WRITEL	Tests character of element connection card and writes element information on to file
XTRAK	Interpret degree of freedom informations
ZZ	Generates OPTIM data which is input by NASTRAN form input cards

1.	Subroutine Name	ADJUST
2.	Purpose	Adjust each NASTRAN form field so that it is either left or right adjusted.
3.	Procedures	Each character of the word is tested to see if it is blank. When a non-blank is met, it is shifted to the end of the word.
4.	Input Arguments	WORD - Input word to be either right or left adjusted Icode - Icode = 8 specifies that the word should be right adjusted - Icode = 1 specifies that the word should be left adjusted
5.	Output Arguments	Word right adjusted, word is stored back into WORD
6.	Error Returns	None
7.	Calling Sequence	CALL ADJUST (Word, ICODE)
8.	Subroutine User	READI
9.	Storage Required	(642 Bytes) 161 words

1.	Subroutine Name:	AONE
2.	Purpose:	Assemble all initialization information for the calculation phase of the program.
3.	Equations and Procedures:	a) Print out title cards. b) Retrieve (I5) coordinate+element data. If requested print this data out. c) Recalculate boundary conditions if symmetry plane nodes are specified. d) Create load matrix from input loads. e) Define individual constraints. f) Call element routines. g) One unit of data I~ written with information for the calculation program.
4.	Input Arguments:	I5 (Unit 13) tape NPOT (Unit 6) printer NSS1 (Unit 1 I/O) N2 No. of nodes NDL No. of Individual Constraints NSYM No. of load conditions ITERN Max. No. of iterations NTCDS No. of title cards
5.	Output Arguments:	C1INP, C2INP Convergence limits NRDF No. of reduced DOF NBOU Total No. of constrained DOF IRST Calculation control
6.	Error Returns:	None
7.	Calling Sequence:	CALL AONE(X,Y,Z,N5,N6,N7,E,AMU,N8,N17,ALL,RHO,SIGU, SIGL,R,N11,N13,N15,NOAL,NOAL2,KL,KL2,LNOD,LNOD2, DISPU,DISPL,NBDF,NBDF2,NSE,DELTA,I5,NPOT,NSS1,NSS2, NSS3,NSS4,IBUKL,IELT,IBK,N1,N2,NAA,NDL,NSG,NNZL, IELI,NREACT,C1INP,C2INP,KNOAL,IBKGP,EM,JMAT,ANGLE)
8.	Input Tapes:	NTAPE (unit 8)
9.	Output Tapes:	NSS1 (unit 1)
10.	Scratch Tapes:	None
11.	Storage Required:	(17840 bytes) 4460 words
12.	Subroutine User:	NEWS
13.	Subroutine Required:	ELEM1 ELEM2 ELEM3 ELEM4 ISOTR
14.	Remarks:	None

1. Subroutine Name: EGMAT

2. Purpose: Generate Eg matrix for triangular and quad plates

3. Equation and Procedures: $Eg = TET^T$

4. Input Argument
   E = 3 x 3 matrix
   T = 3 x 3 transformation matrix

5. Out Arguments Eg = matrix defined by equations above

6. Error Returns None

7. Calling Sequence Call EGMAT (EG, E, T)

8. Subroutine User ELEM3, ELEM4

9. Storage Required: (716 bytes) 179 words

1. Subroutine Name: EMAT

2. Purpose: To generate elasticity matrix for either orthotropic or isotropic properties

3. Equations and Procedures:

Orthotropic (IMAT = 2)

$$E = \begin{vmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{vmatrix}$$

Isotropic (IMAT = 1)

$$E = \begin{vmatrix} E_x & M_{xx} & 0 \\ M_{xy}Ey & Ey & 0 \\ 0 & 0 & G_{xy} \end{vmatrix}$$

4. Input Arguments: E_x, E_y, M_{xy}, G_{xy} - material properties

EM - contains G_{ij} values

IMAT - Orthotropy code = 2 orthotropic
Orthotropy code = 1 isotropic

5. Output Arguments: E - output 3 x 3 matrix defined by equations above

6. Error Returns: None

7. Calling Sequence: Call EMAT (E_x, E_y, M_{xy}, G_{xy}, E, EM, IMAT)

8. Subroutine User: ELEM3, ELEM4

9. Storage Required: (532 bytes) 144 words

1.	Subroutine Name:	ELEM1	
2.	Purpose:	To compute the elemental force and flexibility matrices for an axial force element.	
3.	Input Arguments:	X, Y, Z	Node point coordinates
		E	Modulus of elasticity
		RHO	Density
		N5, N6	Element nodes
		NEL	Element number
		N1	Degrees of freedom/node
		N2	Total number of nodes
4.	Output Arguments:	M	Unreduced direction numbers for degrees of freedom of element
		M(25)	Order of B-Force matrix
		M(26)=1=KKK	Number of force components
		B(1,1)	Start of B-Force matrix
		F(1,1)	Start of flexibility matrix
		WT	Weight of element with unit design variable
		ALEN	Length of element
		ENEL	Modulus of elasticity
5.	Error Returns:	None	
6.	Calling Sequence:	CALL ELEM1 (X,Y,Z,E,AMU,RHO,N7,N5,N6,N8,N9,N10,N11, N12,R,M,NEL,N1,N2,B,F,ALEN,KKK,ENEL,WT)	
7.	Input Tapes:	None	
8.	Output Tapes:	None	
9.	Scratch Tapes:	None	
10.	Storage Required:	(1284 bytes) 321 words	
11.	Subroutine User:	AONE	
12.	Subroutine Required:	None	
13:	Remarks:	The force matrix provides one force component S_1.	

1.	Subroutine Name:	ELEM2	
2.	Purpose:	To compute the element force and flexibility matrices for a shear web element.	
3.	Input Arguments:	X,Y,Z	Node point coordinates
		E	Modulus of Elasticity
		AMU	Poisson's Ratio
		RHO	Element Density
		N5,N6,N7,N8	Element nodes
		NEL	Element number
		N1	Degrees of freedom/node
		N2	Total number of nodes
4.	Output Arguments:	M	Unreduced direction numbers for degrees of freedom element
		M(25)	Rows of force matrix
		M(26)=1=KKK	Number of stress components
		B(1,1)	Start of force matrix
		F(1,1)	Start of flexibility matrix
		WT	Weight of element with unit design variable
		AREA	Surface area of element
		ENEL	Modulus of elasticity
5.	Error Returns:	None	
6.	Calling Sequence:	CALL ELEM2 (X,Y,Z,E,AMU,RHO,N7,N5,N6,N8,N9,N10,N11, N12,R,M,NEL,N1,N2,B,F,AREA,KKK,ENEL,WT)	
7.	Input Tapes:	None	
8.	Output Tapes:	None	
9.	Scratch Tapes:	None	
10.	Storage Required:	(3648 bytes) 912 words	
11.	Subroutine User:	AONE	
12.	Subroutine Required:	None	
13.	Remarks:	A. This element can be defined by 2 or 4 nodes.	
		B. The force matrix provides one force component.	

1.	Subroutine Name:	ELEM3	
2.	Purpose:	To compute the element force and flexibility orthotropic matrices for a triangular plate element, with properties and material angle variation.	
3.	Input Arguments:	X,Y,Z	Node point coordinates
		EE	Modulus elasticity
		AMU	Poisson's ratio
		RHO	Density
		N5,N6,N7	Element nodes
		NEL	Element number
		N1	Degrees of freedom/node
		N2	Total number of nodes
		EM	Material properties
		IMAT	Orthotropic code
		ANGLE	Material angle
4.	Output Arguments:	M	Reduced direction number for degrees of freedom of element
		M(25)	Rows of force matrix
		M(26)=3=KKK	Number of force components
		B(1,1)	Start of matrix force
		F(1,1)	Start of matrix flexibility
		WT	Weight of element with unit design variable
		AREA	Surface area of element
		E	Elasticity matrix (3 x 3)
5.	Error Returns:	None	
6.	Calling Sequence:	CALL ELEM3 (X,Y,Z,EE,AMU,RHO,N7,N5,N6,N8,N9,N10,N11,N12,R,M,NEL,N1,N2,GM,IMAT,ANGLE,B,F,AREA,KKK,E,WT)	
7.	Input Tapes:	None	
8.	Output Tapes:	None	
9.	Scratch Tapes:	None	
10.	Storage Required:	(3510 bytes) 878 words	
11.	Subroutine User:	AONE	
12.	Subroutine Required:	EGMAT,EMAT,MINV,TMAT	
13.	Remarks:	Three components of force at the midpoint are provided.	

1. Subroutine Name: ELEM4

2. Purpose: This routine computes the element force and flexibility matrices for a quadrilateral plate element with property and material angle variation.

3. Input Arguments:

X,Y,Z	Note point coordinates
E	Modulus of elasticity
AMU	Poisson's ratio
RHO	Density
N5,N6,N7,N8	Element nodes
NEL	Element number
N1	Degrees of freedom/node
N2	Total number of nodes
IMAT	Orthotropic code
ANGLE	Material angle
EM	Material properties

4. Output Arguments:

M	Unreduced direction numbers for degrees of freedom of element
M(25)	Rows of force matrix
M(26)=5	Number of force components
B(1,1)	Start of force matrix
F(1,1)	Start of flexibility matrix
AREA	Surface area of element
WT	Weight of element with unit design variable
X_o, Y_o	Coordinates of centroid (local coordinates)
NNE=3	
E	Elasticity matrix order (3 x 3)

5. Error Returns:

6. Calling Sequence: CAL ELEM4 (X,Y,Z,E,AMU,RHO,N7,N5,N6,N8,N9,N10,N11, N12,R,M.NEL,N1,N2,EM,IMAT,ANGLE,B,F,AREA,NNE,E,WT, XC,YC)

7. Input Tapes: None

8. Output Tapes: None

9. Scratch Tapes: None

10. Storage Required: (5688 bytes) 1422 words

11. Subroutine User: AONE

12. Subroutine Required: EGMAT,EMAT,MINV,TMAT

13. Remarks: Five components of force are defined at the midpoint.

1. Subroutine Name: EXIT

2. Purpose: To provide for final stop in the program.

3. Equations and Procedures: Enter and stop.

4. Input Arguments: None

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence: CALL EXIT

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

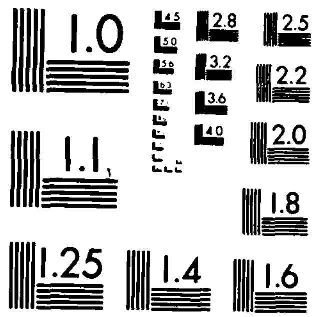
11. Storage Required:

12. Subroutine User: MAIN,OPTIM2

13. Subroutine Required: None

AD-A128 235 FORCE METHOD OPTIMIZATION II VOLUME II USER'S MANUAL 2/3
(U) BELL AEROSPACE TEXTRON BUFFALO NY J R BATT ET AL.
NOV 82 AFWAL-TR-82-3088-VOL-2 F33615-8-C-3214

UNCLASSIFIED F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1. Subroutine Name FOMO

2. Purpose Process force cards

3. Equations and Procedures: The cards are read from file L5 and counted by a counter KFORCE. The input values are (Force 1(i). i = 1, 2), F, and ($F_2(i)$, i = 1, 2, 3); (Force 2(i) = F - $F_n(i)$, i = 1, 2, 3).

4. Input Arguments:
   - OUT = storage for reading data from label field (1st 8 columns)
   - FORCE 1 = set id number, grid point number
   - L5 = input file number
   - L6 = output file number
   - NFO = number of loads

5. Output Arguments: Force 2(i) for each component

6. Error Returns: None

7. Calling Sequence: Call FOMO (FORCE 1, FORCE 2, L5, NFO, K FORCE, OUT, L6)

8. Subroutine User ZZ

9. Storage Required: (712 bytes) 178 words

Note: F is a scale factor for force components read from cards.
FN(i), i = 1, 2, 3 component of force read from cards.

1.	Subroutine Name:	INSPC		
2.	Purpose:	Stores data obtained from input card into working storage.		
3.	Equations and Procedures:	SPCINF	=	Out storage, SPCTYP
4.	Input Argument:	SPCTYP	=	Special code for type of card
		OUT	=	Storage for card data
		NSPCS	=	Maximum number of special input cards
5.	Output Argument	SPCINF		
		NSCARD	=	Counter for no. special input cards.
6.	Error Returns:			
7.	Calling Sequence:	Call INSPC (NSCARD, OUT, NSPCS, SPCTYP, SPCINF)		
8.	Subroutine User:	ZZ		
9.	Storage Required:	(418 bytes) 105 words		

1. Subroutine Name: ISOTR

2. Purpose: Given any two quantities of E, G and μ compute the third if not defined (= 0.0)

3. Equations and Procedures:

$$E = 2G*(1 + \mu)$$
$$G = E*(1 + \mu)/2$$
$$\mu = (E/2G) - 1.0$$

4. Input Arguments:
   - E = Elastic modulus
   - G = Shear modulus
   - μ = Poisson's ratio

5. Output Arguments: See purpose.

6. Error Returns: None

7. Calling Sequence: Call ISOTR (E,NU,G)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (360 bytes) 90 words

12. Subroutine User: AONE

13. Subroutine Required: None

1.	Subroutine Name:	LINK1
2.	Purpose:	Control program flow for the data interpretation and initialization phases.
3.	Equations and Procedures:	Call sort routine which reads input NASTRAN data and adjusts fields and writes on Unit 9. The ZZ routine is called next where the data is interpreted. Next the OPTIM2 routine is activated which controls the initialization phase. Control is then returned to the main routine.
4.	Input Arguments:	W - Dynamic storage NWORK - Maximum size of dynamic storage
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	Call LINK1 (W,NWORK)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(2352 bytes) 588 words
12.	Subroutine User:	Main
13.	Subroutine Required:	SORT, OPTIM2, ZZ, EXIT

1.	Subroutine Name:	MINV
2.	Purpose:	Invert a matrix
3.	Equations and Procedures:	Uses the standard Gauss-Jordan method in which the inverted matrix is stored back on itself.
4.	Input Arguments:	A = Matrix to be inverted N = Order of matrix D = Determinant value L = Work vector of length N M = Work vector of length N
5.	Output Arguments:	A = Contains the inverted matrix
6.	Error Returns:	If D=0, matrix is singular
7.	Calling Sequence:	CALL MINV (A,N,D,L,M)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(1440 bytes) 360 words
12.	Subroutine User:	ELEM3, ELEM4
13.	Subroutine Required:	None
14.	Remarks:	None

1. **Subroutine Name:** NEWS

2. **Purpose:** To determine max core available vs problem size and set up dynamic storage subscripts for the initialization phase of the program.

3. **Equations and Procedures:** Storage requirements for the arrays used by the AONE (utilization routine) are computed from the input parameters.
If the storage required exceeds MAXCOR(the size of the W array) then IER is set to one, a message is written describing the additional storage required and the routine returns to the main routine.
If the storage available is adequate, locations in the W array are computed for the required arrays and routine AONE is called.

4. **Calling Arguments:** (* Input)

*N1	Control for degrees of freedom of a node set to 3 internally.
*N2	No. of nodes in problem.
*ITOT	Number of elements in problem.
*NBOU	Total number of constrained degrees of freedom for a symmetric load condition.
*NSYM	Number of load conditions.
*NDL	Number of individual constraints.
*NTAPE	Input unit for AONE routine (Unit 13).
*NPOT	Print-out data set (Unit 6).
*NSS1	Output data set for initialization phase of program (Unit 1).
*NALD	Total number of load conditions (NSYM).
*ITERN	Maximum number of iterations.
*NTCDS	Number of title cards.
*IREST	Print control for printing input.
IELI	Integer array indicating number of each type element, e.g., IELI(4)=N type 4 elements.
NREACT	Integer array six elements long indicating total number of constraints, e.g. NREACT(1)= number of constrained U components.
*C1INP	Convergence control.
*C2INP	Convergence control.
NRDF	Number of reduced degrees of freedom for summetric load condition.
*IRST	Calculation control which determines use of OPDVIR section.
*MAXCOR	Size of 'W' array available for dynamic storage.
*W	Array used for dynamic storage.
IER	Error control.

5. **Error Returns:** IER≠0 Not enough core available to complete initialization.

6. Calling Sequence: CALL NEWS(N1,N2, NSG,NAAALL,NDL,NAA,NTAPE,NPOT, NSS1,NSS2,NSS3,NSS4,IBUKL,NNZL,IELI,NREACT,C1INP, C2INP,KNLMAX,MAXCOR,W)

7. Input Tapes: None

8. Output Tapes: None

9. Scratch Tapes: None

10. Storage Required: (3078 bytes) 770 words

11. Subroutine User: OPTIM2

12. Subroutine Required: AONE

13. Remarks: None

1.	Subroutine Name:	OPINPT	
2.	Purpose:	To complete NTAPE (Unit 13); input to the AONE routine.	
3.	Equations and Procedures:	Process input needed by the AONE routine in this order. COORD ELEM OEXTERN LINKS BOUND OLOADS ICON GCON OPDVIR Place all information on NTAPE (Unit 13). Print diagnostic messages if necessary.	
4.	Input Arguments:	NPIT	Unit 5 (card reader)
		NPOT	Unit 6 (printer)
		NTAPE	Unit 13 (tape)
		N2	Number of nodes
		NDL	Number of individual constraints
		NNZL	Number of input cards for loads
		NSS1	Unit 1 (tape)
		NREF NGRD NDOFPN NALD ITOT NIBCP	Info from System section input
5.	Output Arguments:	NREACT	Array indicating number of bounded DOF.
		IER	Error indicator.
6.	Error Returns:	IER=0	
7.	Calling Sequence:	CALL OPINPT(NPIT,NPOT,NTAPE,NREF,NGRD,NDOFPN,NALD, ITOT,NIBCP,N1,N2,NSGIN,NDL,NAA,NDLA,NREACT,NNZL,IELI, IRST,NAANUM,NSG,NSS1,KNLMAX,IER,GRID,IBOUND,NBOUND, INODE,NBDF,DISPU,DISPL,LINKB,LINKN,ILINK,DVIR,IBUCKL, NAM,IBO,IG1,IG2,IG3,IBGP)	
8.	Input Tapes:	NSS1	
9.	Output Tapes:	NTAPE	
10.	Scratch Tapes:	NSSL (Unit 1) used to merge ELEM and OEXTRN sections.	
11.	Storage Required:	(14922 bytes) 3731 words	
12.	Subroutine User:	OPTIM2	

13. Subroutine Required: None

14. Remarks: None

1.	Subroutine Name:	OPTIM2
2.	Purpose:	To control the program execution for phase 2.
3.	Equations and Procedures:	Identify all external files and rewind them. Read REPORT, TITLE, SYSTEM and OPTIM sections to define variables needed for dynamic storage. If there are any title cards read them and put them on NTAPE (Unit 8). Print message if any sections are out of order. Call OPINPT routine to read rest of input sections. If there is any error (IER≠0) call exit. Call NEWS routine to perform dynamic storage allocation and call routines to perform initialization. If there is any error (IER≠0) call exit.
4.	Input Arguments:	None
5.	Output Arguments:	None
6.	Error Returns:	(IER≠0) call exit.
7.	Calling Sequence:	Call OPTIM2 (WORK, NPIT)
8.	Input Tapes:	WORK = Work storage NPIT = Input file number
9.	Output Tapes:	NTAPE (Unit 13)
10.	Scratch Tapes:	NSS1 (Unit 1)
11.	Storage Required:	(3784 bytes) 946 words. The work array is used to dynamically locate arrays.
12.	Subroutine User:	LINK1
13.	Subroutine Required:	OPINPT NEWS EXIT
14.	Remarks:	None

1.	Subroutine Name	READI
2.	Purpose:	Reads and modifies input data.
3.	Equations and Procedures:	Reads the card data from file J5, right adjust the data fields, counts each data type, prints the data and finally stores the modified data on file J6.
4.	Input Argument:	LABEL = array of BCD label codes ILAB = array of label integers ISPECL = array of special labels NILAB = total no. of labels J6 = output file number NSPECL = total no. special labels.
5.	Output Arguments:	NCARDS = array of card counters L7CASE = code to indicate that file is = 7.
6.	Error Returns:	None
7.	Calling Sequence:	CALL READI(LABEL,ILAB,ISPECL,NCARDS,NILAB,NSPECL,J6,L7CASE)
8.	Subroutine User:	SORT
9.	Subroutines Used:	ADJUST
10.	Storage required:	(1940 bytes) 485 words

1.	*Subroutine Name:*	SORT
2.	Purpose:	Sort and count data based on LABEL information.
3.	Equations and Procedures:	Using READI, the data is read and counted. The final counters are then modified.
4.	Input Arguments:	L4 = file number for storage of sorted input deck.
5.	Output Arguments:	MEL Total no. Elements MGR Total no. Grid points MMT Total no. Materials MOGCON Total no. Generalized Constraints MICON8 Total no. Individual Constraints MILINKS Total no. Links MFO Total no. Forces MMO Total no. Moments MLOADS Total no. Loads
6.	Error Returns:	None
7.	Calling Sequence:	Call SORT(MEL,MGR,MMT,MOGCON,MICON8,MILINKS,MFO, MMO,MLOADS,MGROUP,MSPCS,L4,L7CASE)
8.	Subroutine User:	LINK1
9.	Subroutines Used	READI
10.	Storage Required:	(1578 bytes) 395 words

1.	Subroutine Name:	SPCSUB	
2.	Purpose:	Process SPC (single point constraint) cards.	
3.	Equations and Procedures:	Boundary information is processed as read in OUT (I) and NOUT (I). This information is interpreted and stored in LBOUND.	
4.	Input Arguments:	NOUT	Input data storage
		OUT	Input data storage
		NSPC	No. SPC cards
		KWORD	Work storage
		NUM	Work storage
		NKIND	Type of boundary information available
		NGR	Total no. grid points
5.	Output Arguments:	LBOUND	Boundary array information
		NBCARD	Counter of boundary information
6.	Error Returns:	None	
7.	Calling Sequence:	Call SPCSUB (NOUT,OUT,NSPC,KWORD,NUM,LBOUND,NBCARD, NKIND,NGR)	
8.	Subroutine User:	ZZ	
9.	Subroutines Used:	XTRAK	
10.	Storage Required:	(1602 bytes) 401 words	

1. Subroutine Name: TMAT

2. Purpose: Generate transformation matrix for triangular plate orthotropic material angle.

3. Equations and Procedures:

$$T = \begin{bmatrix} \cos^2\theta & \sin^2\theta & \sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -\sin\theta\cos\theta \\ -2\sin\theta\cos\theta & 2\sin\theta\cos\theta & \cos^2\theta-\sin^2\theta \end{bmatrix}$$

4. Input Arguments: THETA = material angle

5. Output Arguments: T = 3 x 3 matrix

6. Error Returns: None

7. Calling Sequence: CALL TMAT (THETA,T)

8. Subroutine User: ELEM3, ELEM4

9. Subroutines Used: None

10. Storage Required: (420 bytes) 105 words

1. Subroutine Name: TMAT

2. Purpose: Generate transformation matrix for triangular plate orthotropic material angle.

3. Equations and Procedures:

$$T = \begin{bmatrix} \cos^2\theta & \sin^2\theta & \sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & -\sin\theta\cos\theta \\ -2\sin\theta\cos\theta & 2\sin\theta\cos\theta & \cos^2\theta-\sin^2\theta \end{bmatrix}$$

4. Input Arguments: THETA = material angle

5. Output Arguments: T = 3 x 3 matrix

6. Error Returns: None

7. Calling Sequence: CALL TMAT (THETA,T)

8. Subroutine User: ELEM3, ELEM4

9. Subroutines Used: None

10. Storage Required: (420 bytes) 105 words

1. Subroutine Name: WRITEL

2. Purpose: Tests character of element connection card and writes element information into file.

3. Equations and Procedures: If C_1 and C_3 are 0, C_2 and C_4 are stored. If C_2 and C_4 are 0, C_1 and C_3 are stored.

4. Input Arguments:

L7	Output file number
IGR	Grid point number
M	Position of A array to be restored.
C_1, C_2, C_3, C_4	Input codes

5. Output Arguments: IGR and A array are stored on file L7.

6. Error Returns: None

7. Calling Sequence: CALL WRITEL (C_1,C_2,C_3,C_4,M,IGR,L7)

8. Subroutine User: ZZ

9. Surboutines Used: None

10. Storage Required: (578 bytes) 145 words

1.	Subroutine Name:	XTRAK	
2.	Purpose:	Interpret degree of freedom information	
3.	Equations and Procedures:	Interprets NWORD and breaks this down into 6 individual components. These components are then stored in KWORD array.	
4.	Input Arguments:	NWORD	No. components input word
		NP	Control word
5.	Output Arguments:	KWORD	Output data array
		NUM	Total no. DOF recognized
6.	Error Returns:	None	
7.	Calling Sequence:	CALL XTRAK (NWORD,KWORD,NUM,NP)	
8.	Subroutine User:	SPCSUB,ZZ	
9.	Subroutines Used:	None	
10.	Storage Required:	(714 bytes) 179 words	

1. Subroutine Name: ZZ

2. Purpose: Generates OPTIM data which is input by NASTRAN format input cards.

3. Equations and Procedures: Each card is read and interpreted based on content and use in the OPTIM program.

4. Input Arguments: L5 = Input file tape number
L7 = Output file tape number

5. Output Arguments: All of the grid point, boundary condition, element, material property, load, constraint and buckling information arrays needed by OPTIM.

6. Error Returns:

7. Calling Sequence: CALL ZZ (NAST,NOPT,MATNO,NPROP,NBUCK,NNODES,NOID, REF,OPDVIR,LBOUND,COOR,AMAT,MID,EYEC,LINKS,NEL,NGR, NMT,NICON8,NLINKS,NOGCON,NP,FA,NAP,GA,NFO,NMO,NLOADS, NSPCS,IGRID,SPCINF,GCOND,FORCE3,ANGLE,FORCE1,FORCE2, MOMNT1,MOMNT2,OPLOAD,GROUP,L5,L7).

8. Subroutine User: LINK1

9. Surbroutines Used: FOMO
INSPC
SPCSUB
WRITEL
XTRAK

10. Storage Required: (18018 bytes) 4505 words

A.4.2 LINK2 SUBPROGRAMS

Each subprogram of LINK2 is described preceded by a glossary of argument definitions.

A.4.2.1 BRIEF SUBROUTINE DESCRIPTIONS

Subroutine Name	PURPOSE
AA	Structure cutter control stage
ASSEM	Compute row and column vectors for matrix assembly
BASIC	Statics analysis outputs X, S, Delta and Y
BASICD	Dynamic analysis outputs X, P, Delta and mass matrix infor
DYNERR	Compares storage required against storage allotted
EIG	Generates eigenvalues and eigenvectors by power method
EQSOL	Compute Lagrange multipliers for "on" constraints
FSD	Fully stressed design loops on BASIC routine
GELS	Solves a system of simultaneous linear equations with symmetric coefficient matrix
INTPR	Auxiliary subroutine for linear programming ITER
ITER	Linear programming subroutine
LINK2	Main routine for structure cutter and calculation phases
LOC	Auxiliary routine for MPRD
MAXVAL	Generate maximum value of eigenvector
MFGR	Structure cutter routine
MFSD	Factor a given symmetric positive matrix
MINV2	Symmetric matrix inversion calling routine
MPRD	General matrix product

Subroutine Name	Purpose
MRPRNT	Matrix print routine
MSTR	Change storage mode of a matrix
OPTFR	Control program for optimization
PIVOT	Performs a simplex pivot about a given matrix element
PRIN1	Prints structure design variables and computes total weight
REGEND	Regenerates dynamic X_K and P_K from old Delta-K
SBTFD	General triple matrix product $(B^T F D)$
SB1IR	Extract element and reaction B matrices from structure cutter matrix
SD1IR	Extract element and reaction D matrices " " " "
SHIFTA	Shift dynamic storage locations and initialize matrix
SIMQ	Obtain solution of a set of simultaneous linear equations
SINV	Invert a given symmetric positive definitive matrix
STDB	Computes element forces and reactions
SUBA	Generates Ao^{-1} from structure cutter matrix
S222	Compute BASIC structural matrices for dynamic analysis
S231	Generates element forces as function of redundant X and load P
S231A	Computes start and final columns of element matrix
S241	Generates ■ bar matrices for each element
S311	Assembles bar matrices into PHI PSI and Omega matrices
S311D	Compute PHI^{-1} and PSI
S314	Solves for $X = -\phi^{-1}\psi P$ (redundants)
S315	Solves for displacements $\Delta = \psi^T x + \Omega P$
S316	Global flexibility matrix $F = \Omega - \psi^T \phi^{-1}\psi$

Subroutine Name	Purpose
S321	Dynamic eigenvalue matrix $Q = F_{ij} (M_i M_j)^{1/2}$
S323	Inertial load mode $P_K{}^i = M_i^{1/2} q_K{}^i$
S324	Displacement mode $\Delta_K{}^i = P_K{}^i/\omega_K{}^2 M_i$
S324R	Solves for P_K from Delta-K for dynamic reanalysis
S4310	Generates partial $g\sigma$ with respect to X
S4310A	Output calculated matrix element to a temporary file
S4310B	Inserts calculated element into analysis matrix
S432	Generates partial of g_A with respect to A
S4323	Generates partial of $g\Delta$ with respect to A
S4325	Generates frequency constraint g_ω
S4325A	Generates f_i for g_ω and both partials of g_ω
S433	Generates y_i = Mases - Hencky criteria stress
S435	Generates partial of g_σ with respect to A
S451	Partial Newton Raphson
S461	Full Newton Raphson
S4614	Second partials with respect to AJ, AK
S4621	Calculates second partial with respect to A_J, X_k
S4622	Second partial of Lagrangian with respect X_j, X_k
S4622A	Auxiliary routine for second partial of Y WRT S for triangles and quads
S466	Calculate second partial of g with respect to A_j and X_K
TAPE11	Generate files I11 and structure cutter initialization

Subroutine Name	Purpose
TESTR	Test for divergence in design
TRPRNT	Transpose matrix print

A.4.2.2 GLOSSARY OF ARGUMENT DEFINITIONS FOR LINK2 ROUTINES

A — Vector containing the current design parameters (length = NE)

ALAMBD — Vector for storage of Lagrange multipliers (length - NE + NSNL+NDCNL+NXNL+NWNL)

AMIN — Vector for storage of the minimum permissible design variables (length = NE)

AREA — Vector for storage of surface areas of quad, triangle and shear panel type elements (length = NE)

COND — Input convergence criteria ($\left|\frac{A_{old} - A_{new}}{A_{old}}\right| \leq$COND)

IAREA — Vector identifying unknown design variables and redundants. This vector is used for row or column labels when assembling the partials into the coefficient matrix for solution of a set of linear equations (length = NE+NXNL); IAREA(I) = 0; Variable known - IAREA(I)>0; Variable unknown and Value = Label

IC — Dummy vector containing input matrix column labels

ICC — Iteration counter

ICHECK — Vector for intermediate output, identifying the convergence status of unknowns, 0 = has not converged, 1 = has converged
2 = not an unknown
(Length=NE+NSNL+NDCNL+NXNL+NWNL)

IDEL — Pointer to starting position in vector W at which the displacements are stored

IDYN — Switch for inclusion of dynamic constraints

0 - No dynamic constraints in problem
1 - Yes dynamic constraints in problem, but not yet included
2 - Yes dynamic constraints in problem

IMUAL — Pointer to starting position in vector IMU for storage of design constraint "active/not active" switches

IMUSL — Pointer to starting position in vector IMU for storage of stress constraint "active/not active" switches

IMUDL — Pointer to starting position in vector IMU for storage of displacement constraint "active/not active" switches

IMUXL — Pointer to starting position in vector IMU for storage of redundants "active/not active" switches

IMUWL	Pointer to starting position in vector IMU for storage of frequency constraint "active/not active" switches
IMU	Vector identifying "active" constraints. This vector is used for column or row labels when assembling the partials into the coefficient matrix for solution of a set of linear equations (length=NE+NSNL+NDCNL+NXNL+NWNL) IMU(I) = 0; constraint is not active IMU(I) > 0; constraint active, value of IMU(I) = Label
IRST	Control for execution of statics, dynamics or optimization
ISTRES	Stress-ratio method control variable
IR	Dummy vector containing input matrix row labels
ISI	Pointer to starting position in vector "W" at which the S values are stored
IYI	Pointer to starting position in vector "W" at which the Y values are stored
IX	Pointer to starting position in vector "W" at which the redundants are stored
IPK	Pointer to starting position in vector "WDYN" at which the P_K values are stored
IXK	Pointer to starting position in vector "WDYN" at which the X_K values are stored
IZ	Pointer to starting position in vector "WDYN" at which the mass matrix is stored
KL	Vector containing unreduced degrees of freedom numbers for reactions (length = NBOU)
LCOL	Initial column number of matrix waray for assembly of a group of coefficients. Used by routine assem to load the column label vector
LROW	Initial row number of matrix waray for assembly of a group of coefficients. Used by routine assem to load the row label vector
LOW	Vector containing lower displacement limits (length = NDC)
MAXIT	Maximum permissible iterations to be performed in all convergence loops
NASTAR	The number of active ('on') design variable (minimum size) constraints

NBOU	Length of vector KL
NC	Dummy argument indicating total number of columns in the coefficient matrix.
NCOL	Number of columns in the A-matrix from the structure cutter routine NCOL = NSE+NBOU+1
NCOL1	Number of columns in matrix waray into which a group of coefficients is to be assembled. Used by ASSEM routine to load column label vector
ND	Vector containing degree of freedom for displacement constraints (length = NDC)
NDC	Number of displacement constraints
NDCNL	Number of displacement constraint variables = NDC x number of loads
NDN	Number of reduced degrees of freedom
NDNNL	NDN x number of loads
NDSTAR	The number of active ('on') displacement constraints
NDT	Total number of degrees of freedom in problem
NDTNX	Total number of columns in structure cutter matrix
NE	Number of elements in structure
NLL	Number of load conditions
NODE	Total number of nodes in structure
NR	Dummy argument containing total number of rows in the coefficient matrix
NROW	Number of rows in the A-matrix from structure cutter routine
NROW1	Number of rows in matrix WARAY into which a group of coefficient is to be assembled. Used by ASSEM routine to load row label vector
NSE	Number of element forces in the structure
NSENL	NSE x number of loads
NSET	Dummy vector for identifying the active constraints as determined by the FSD routine (length = NSNL)
NSNL	Number of stress constraint variables = NE x number of loads

NSSTAR	The number of active ('on') stress constraints
NTRANS	Control to transpose output of a partial computation routine prior to assembly in matrix waray. 1 = no transpose 2 = transpose -N = assemble in lower triangle
NTYPE	Vector identifying element type (length = NE)
NW	Number of modes
NWNL	Number of frequency constraint variables
NWSTAR	Number of active ('on') frequency constraints
NX	Number of redundants
NXNL	Number of redundant constraint variables = NX x number of loads
NXSTAR	Number of active ('on') redundant constraints
OMEGA	Assembled structural matrix
P	Vector containing static loading of structure (length = NBOU)
PHI	Assembled structural matrix
PRD	Control for printing of displacements every iteration
PRR	Control for printing of reactions every iteration
PRS	Control for printing of stress every iteration
PRI	Control for printing of intermediate data
PSI	Assembled structural matrix
S	Vector containing parameters describing stress state of elements (length = NSENL)
SIG	Vector containing stress limits (length = NE)
UP	Vector containing upper displacement limits (length = NDC)
W	Vector containing all redundants, displacements, S and Y values. (length = NXNL+NDNNL+NSENL+NSNL)
WARAY	Dynamic storage. Vector for storage of computed partials and any work storage required by Phase 4 subroutines.
WDYN	Vector for storing dynamic variables DEL, P_K, X_K and Z. (Length = 2xNDN*NWNL+NX*NWNL+NE*NDN)

WS	Frequency limit
WT	Vector containing element normalized weight (length = NE)
X	Vector containing redundants (length = NXNL)
XC	Vector containing the X-coordinates of centroid of quad elements (length = NE)
Y	Vector containing element forces (length = NSNL)
YC	Vector containing the Y-coordinates of centroid of quad elements (length = NE)

1. Subroutine Name: AA

2. Purpose: Structure cutter control stage

3. Equations and Procedures: The de bug print control determines whether the matrix to be processed is printed before and after the structure cutter routine MFGR.

4. Input Arguments: A,M,N,LL,AO,U,WORK, WORK1,BO,IROW,ICOL

   These arguments reserve storage for internal calculation by structure cutter routines AONE and MFGR.

5. Output Arguments: A,IROW,ICOL are returned after the structure cutter process and written on FILE I12 by the TAPEII routine.

6. Error Returns: If the rank of the matrix does not equal the no. of rows an error message is printed and the routine returns to TAPEII.

7. Calling Sequence: Call AA(A,M,N,CC,AO,U,WORK,WORK1,BO,IROW,ICOL)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1314 bytes) 425 words

12. Subroutine User: TAPEII

13. Subroutine Required: SUBA,MFGR,MRPRNT

1.	Subroutine Name:	ASSEM
2.	Purpose:	Compute row and column vectors for matrix assembly.
3.	Equations and Procedures:	a) Load row vector with row numbers (sequential) for assembly of an input matrix. b) Load column vector with column numbers (sequential) for assembly of an input matrix.
4.	Input Arguments:	LROW,LCOL,NROW1,NCOL1,IR,IC See Glossary for definitions.
5.	Output Arguments:	IR,IC See Glossary for definitions
6.	Error Returns:	None
7.	Calling Sequence:	Call ASSEM (LROW,LCOL,NROW1,NCOL1,IR,IC)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(442 BYTES) 111 words
12.	Subroutine User:	OPTFR,S451,S461,EQSOL
13.	Subroutine Required:	None

1. Subroutine Name: BASIC

2. Purpose: Statics analysis computes X,S,Y and Delta

3. Equations and Procedures:
   a) Compute working storage requirements
   b) Compute PHI inverse, PSI and Omega. Call S311.
   c) Compute redundants if requested. Call S314.
   d) Compute displacements (call S315) and print if requested.
   e) Compute work storage requirements.
   f) Read row labels, col labels and A-matrix from structure cutter output.
   g) Compute S(I). Call S231.
   h) Compute stress and 1/(I) for each element. Call S433.
   i) Compute reactions if requested. Call S231.
   j) Print reactions if requested.

4. Input Arguments: PRD,PRS,PRR,NTYPE,AMIN,WT,AREA,XC,YC,A,KL,P,NE, NDN,NX,NBOU,NSE,NODE,NDT,NDTNX,W
   See Glossary for definitions.

5. Output Arguments: NCOL,W
   See Glossary for definitions.

6. Error Returns: None

7. Calling Sequence: Call BASIC (PRD,PRS,PRR,NTYPE,AMIN,WT,AREA,XC, YC,A,KL,P,NE,NDN,NX,NBOU,NSE,NODE,NDT,NDTNX,NCOL,

8. Input Tapes: NTAPE, UNIT 12 (output of structure cutter)

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (3770 BYTES) 943 words

12. Subroutine User: FSD,OPTFR,S451,S461

13. Subroutine Required: S314,S315,S231,S433,S311

1. Subroutine Name: BASICD

2. Purpose: To perform a dynamic analysis and return data that will be used in a dynamically constrained optimization problem.

3. Equations and Procedures: S222 is called and from Tape I13 and the element design parameters, matrices ϕ, ψ, Ω and M are formed. Routine S316 is then called to form the flexibility matrix. Routine S321 then replaces the flexibility matrix by the eigenvalue matrix. The EIG routine then solves for the eigenvalues and eigenvectors. The inertial load modes PK are computed in routine, S323. The displacement modes are computed by S324. The mode shapes are normalized on the max component by MAXVAL and printed. The redundant modes XK are computed by S314. If IPRT=1 (set at the beginning of the routine) the routine returns, otherwise the inertial stress and inertial reaction calculation and print is done by S231 and S433.

4. Input Arguments: PRD,PRS,PRR,NTYPE,AREA,XC,YC,A,KL,NE,NDN,NX, NBOU,NSE,NODE,NDT,NDTNX,NCOL,NW
   See Glossary for definition.
   W - working storage
   NOI - maximum number of iterations
   CRIT - convergence criteria

5. Output Arguments: In the W1 array DEL(NDN,NW),PK(NDN,NW), XK(NX,NW), Z(NE,NDN) are contained.

6. Error Returns: None

7. Calling Sequence: Call BASICD(PRD,PRS,PRR,NTYPE,AREA,XC,YC,A,KL,NE, NDN,NX,NBOU,NSE,NODE,NDT,NDTHX,NCOL,W1,W,NW, NOIT,CRIT)

8. Input Tapes: I12

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (4526 BYTES) 1142 words

12. Subroutine User: OPTFR

13. Subroutine Required: EIG, S222,S231,S314,S316,S321,S323,S324,S433, MAXVAL,MRPRNT

1.	Subroutine Name:	DYNERR
2.	Purpose:	Compares storage required against storage allotted.
3.	Equations and Procedures:	a) Compute total storage required. b) Call EXIT if required storage greater than that dimensioned in LINK2.
4.	Input Arguments:	NEND - Previous storage requirement NEND1 - Additional storage requirement
5.	Output Arguments:	IERR
6.	Error Returns:	Writes dynamic storage allocation error... call EXIT.
7.	Calling Sequence:	Call DYNERR (NEND,NEND1,IERR)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(538 BYTES) 135 words
12.	Subroutine User:	LINK2
13.	Subroutine Required:	EXIT

1. Subroutine Name: EIG

2. Purpose: Generates eigenvalues and eigenvectors by power method.

3. Equations and Procedures:

$$[A]\left[\,\right]_{X_n} = \lambda\left[\,\right]_{X_{n+1}}$$

This equation is repeated until X_n and X_{n+1} fall within the criteria and λ_n and λ_{n+1} converge.

The matrix is deflated removing the higher eigenvalue and then the previous iteration is repeated to get the next value and vector.

4. Input Arguments:

- A - Eigenmatrix less than OR = (N94,N94)
- N - Order of matrix A less than OR = N94
- IPRINT - Print iteration control
- NEIG - No. of eigenvalues requested less than OR=N20
- GUESS - Guess vectors (N94,N20) usually 1.0
- CRIT - If =0 then .001 relative error criteria
- NOIT - If =0 then 54 maximum no. of iterations
- NEL - Selects IST eigenvalue requested usually =1
- N94 - Order of a matrix
- N20 - No. of eigenvalues
- I6 - Print output unit
- XIN,XI,XIMIN,XIP,XIMINP,XINP-storage for successive eigenvector iterations

5. Output Arguments: ICOUNT = Total No. of eigenvalues found
VECTOR = Eigenvectors ROOT = Eigenvalues

6. Error Returns: NERR = 0 NO ERROR = 1 eigencols Do not converge = 2 Eigenrows do not converge = 3 valves do not converge

7. Calling Sequence: Call EIG(A,N,IPRINT,NEIG,GUESS,ROOTS,VECTOR, NERR,CRIT,NOIT,ICOUNT,NEL,N94,N20,I6, XIN, XI,XIMIN,XIP,XIMINP,XINP)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (4438 BYTES) 1110 words

12. Subroutine User: BASICD

13. Subroutine Required: None

1.	Subroutine Name:	EQSOL
2.	Purpose:	Compute Lagrange multipliers for "on" constraints.
3.	Equations and Procedures:	a) Compute matrix dimensions and initialize arrays,WARAY & IAREA. b) Compute and assemble the partials G*A/A, G*S/A,G*S/X,G*D/A,G*D/X,GX/A,GX/X and G*W/A in array WARY by calling routines ASSEM,S432,S435,S4310,S4323 and S4325. c) Solve linear equations for Lagrange multipliers. Call SIMQ. d) Store multipliers in vector ALAMBD. e) Reset vector IAREA.
4.	Input Arguments:	W,WARAY,IAREA,IMU,NASTAR,NSSTAR,NDSTAR,NX, NWSTAR,IMUAL,IMUSL,IMUDL,IMUXL,IMUWL,AMIN,A,NE, IYI,ISI,NTYPE,SIC,AREA,XC,YC,NSE,NDT,NCOL,NOTNX, NBOU,NDC,P,NDN,UP,LOW,ND,IX,NW,WT,NSNL,NDCNL, NWNL,WDYN,PRI,WS,IPK,IXK,IZ See Glossary for definitions.
5.	Output Arguments:	ALAMBD - See Glossary
6.	Error Returns:	Write EQSOL matrix is singular..STOP.
7.	Calling Sequence:	Call EQSOL(ALAMBD,W,WARAY,IAREA,IMU,NASTAR, NSSTAR,NDSTAR,NX,NWSTAR,IMUAL,IMUSL,IMUDL, IMUXL,IMUWL,AMIN,A,NE,IYI,ISI,NTYPE,SIG,AREA, XC,YC,NSE,NDT,NCOL,NDTNX,NBOU,NDC,P,NDN,UP,LOW, ND,IX,NW,WT,IERR,NSNL,NDCNL,NXNL,NWNL,WDYN,PRI, WS,IPK,IXK,IZ,ARAY)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(4516 BYTES) 1137 words
12.	Subroutine User:	OPTFR
13.	Subroutine Required:	SIMQ,S432,S435,ASSEM,S4310,S4323,S4325,MRPRNT, SHIFTA,S4310B

1. Subroutine Name: EXIT

2. Purpose: To provide one stop in the program.

3. Equations and Procedures: This routine returns the program to the system monitor. (Same as LINK1)

4. Input Arguments: None

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence: Call EXIT

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required:

12. Subroutine User: DYNERR

13. Subroutine Required: None

1.	Subroutine Name:	FSD
2.	Purpose:	Fully stressed design - loops on BASIC routine.
3.	Equations and Procedures:	a) Initilize multiple load variables and iteration counter b) Convergence loop 1. Perform a statics analysis based on new design (compute Y(I) (Call BASIC) 2. For each design variable, compute a new design for each load condition and store the largest value and the load condition number ANEN=Y(I)/SIG(I) 3. Set design variable to minimum value if smaller than minimum. 4. Check for convergence and store new design. 5. Reset the stress constraint vector based on load condition number. 6. If not all design variables converged and iteration number is less than maximum, repeat convergence loop. c) Reset stress constraint vector to 0 for any design which is minimum. d) Compute structure weight.
4.	Input Arguments:	PRD,PRS,PRR,NTYPE,AMIN,WT,AREA,XC,YC,A,SIG,KL,P, NE,NDN,NX,NBOU,NSE,NODE,NDT,NDTNX,W,NSET,COND, MAXIT (See Glossary for definitions)
5.	Output Arguments:	A,W - See Glossary for definitions NSET - Vector containing active constraint identification
6.	Error Returns:	None
7.	Calling Sequence:	Call FSD(PRD,PRS,PRR,NTYPE,AMIN,WT,AREA,XC,YC,A, SIG,KL,P,NE,NDN,NX,NBOU,NSE,NODE,NDT,NDTNX,W, NSET,COND,MAXIT)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(2766 BYTES) 692 words
12.	Subroutine User.	OPTFR
13.	Subroutine Required:	BASIC,TRPRNT

1.	Subroutine Name:	GELS
2.	Purpose:	To solve a system of simultaneous linear equations with symmetric coefficient matrix.
3.	Equations and Procedures:	Solution is obtained by means of Gauss-elimination with pivoting in main diagonal, in order to preserve symmetry in remaining coefficient matrix. a) Search for greatest main diagonal element. b) Elimination loop. 1) Test usefulness of symmetric algorithm 2) Pivot, row reduction, and row interchange in right-hand side matrix 3) Row and column interchange and pivot row reduction in coefficient matrix 4) Save column interchange information 5) Element reduction and search for next pivot c) Back substitution and row interchange
4.	Input Arguments:	R - M by N right-hand side matrix A - Upper triangular part of the symmetric M by N coefficient matrix M - The number of equations N - The number of right-hand side vectors EPS - Relative tolerance for test on loss of significance AUX - M-1 auxiliary storage vector
5.	Output Arguments:	R - M by N matrix containing solution to equations
6.	Error Returns:	IER
7.	Calling Sequence:	Call GELS(R,A,M,N,EPS,IER,AUX)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	1730 bytes (433 words)
12.	Subroutine User:	S461
13.	Subroutine Required:	None

1.	Subroutine Name	INTPR
2.	Purpose:	Prints linear programming tableau for routine ITER
3.	Equations and Procedures:	a) Print vector LABC b) Print vector LABR and ARRAY "A"
4.	Input Arguments:	A - Array of dimension MM*N to be printed LABR - Vector of length M to be printed (row labels) LABC - Vector of length N to be printed (column labels) MM - First dimension of "A" in calling program M - Number of rows in ARRAY "A" N - Number of columns in ARRAY "A" IT - Number of iterations
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	Call INTPR(A,LABR,LABC,MM,M,N,IT)
8.	Input Tapes:	None
9.	Output File:	Unit 101 (printer)
10.	Scratch Tapes:	None
11.	Storage Required:	(656 BYTES) 164 words
12.	Subroutine User:	ITER
13.	Subroutine Required:	None

1. Subroutine Name: ITER

2. Purpose: Solve linear equations with provision for the automatic treatment of any or all of the following: Equality constraints, greater than or equal constraints, less than or equal constraints, positive variables, free variables and either maximization or minimization of the objective function.

3. Equations and Procedures: Method

The method is a primal-dual algorithm (sometimes referred to as the 'criss-cross' method). The condensed tableau is pivoted in the following manner. First artificial variables (which correspond only to equality constraints) are pivoted out of the basis and free variables (if any) are pivoted in. If there are still artificial variables in the basis after all free variables have been pivoted in, then the remaining artificial variables are pivoted out and positive variables are pivoted in. If there are more free variables than artificial, then the remaining non-basic free variables are next pivoted in and positive variables out. Now alternate dual and primal iterations are taken until the solution becomes primal or dual feasible. From this point on only primal or only dual iterations are taken until the optimal solution is attained.

In the event the solution is primal feasible but not dual feasible and no primal iteration may be taken, the solution is primal unbounded. Similarly, if the solution is dual feasible but not primal feasible and no dual iteration may be taken, the solution is dual unbounded. If the solution is neither primal nor dual feasible, and neither a primal nor a dual iteration may be taken, then the solution is either unbounded or infeasible in either the primal or the dual sense.

For reference see, 'Linear and Integer Programming,' by Stanley Zionts, Prentice-Hall, 1974.

4. Input Arguments: A Array of dimension MM X N

A(1,2)-A(1,N) Contain the coefficients of the cost vector.

A(2,1)-A(M,1) Contain the RHS's of the constraint equations.

Rows 2-M contain, starting in column 2, the coefficients of the constraint equations.

A(1,1) is irrelevant

	LABR	Vector of length M containing row labels LABR(I) indicates the type of constraint corresponding to row I as follows: -1=LE, 0=Equality, 1=GE LABR(1) is irrelevant
	LABC	Vector of length N containing column labels LABC(I) indicates the type of activity variable corresponding to column I of array A as follows: 1-positive variable, 0=free variable LABC(1) is irrelevant
	MM	First dimension of array "A" in calling program. If "A" is singly subscripted MM=M
	M	Number of constraints +1
	N	Number of activity variables +1
	NMAX	Code to maximize (1), or minimize (-1)
	TOL	Minimum permissible absolute value for non-zero elements of array "A". Elements falling below TOL are set to 0.0
	TOL1	Minimum permissible absolute value for an element to be used as a pivot
	IPFLAG	Print code for array "A" -3 = no printout -2 = print after final iteration -1 = print before initial and after final iteration 0 = print before initial iteration 1 = print before initial iteration and after all succeeding iterations
5. Output Arguments:	A	Array of dimension MMX N (A(1,1) x MMAX) is the optimal value of the cost function A(2,1)-A(M,1) contain corresponding values of basic values
	LABR	Vector of length M containing row labels LABR(1)=0 The value of activity variable number (ABS(LABR(I)) is contained in A(I,1)
	IR	Number of iterations

6. Error Returns: KOPT

7. Calling Sequences: Call ITER (A,LABR,LABC,TOL,TOL1,MM,M,N,MMAX,IPLAG, IT,KOPT)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: 2778 bytes (695 words)

12. Subroutine User: OPTFR

13. Subroutine Required: PIVOT INTPR

1.	Subroutine Name:	LINK2
2.	Purpose:	Main calling routine for structure cutter and calculation phases.
3.	Equations and Procedures:	a) Initialize I/O unit variables. b) Compute dynamic storage requirements and check against maximum allocated (call DYNERR) for TAPEII routine. c) Read phase 2 output and execute structure cutter routines call TAPEII. d) Compute dynamic storage requirements for storage of phase 3 output and check against maximum allocated, call DYNERR. e) Read phase 3 output - from file I11. f) Initialize optimization variables. g) Compute dynamic storage rquirements for phase 4 and check against maximum allocated, call DYNERR. h) Execute phase 4 computations (statics or dynamics or optimization), call OPTFR.
4.	Input Arguments:	W dynamic storage NWORK maximum size of dynamic storage
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	Call LINK2 (W,NWORK,AAA)
8.	Input Tapes:	NSS2 Unit 1 I11 Unit 11
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(4468 bytes) 1117 words
12.	Subroutine User:	MAIN
13.	Subroutine Required:	DYNERR, TAPE11, OPTFR

1.	Subroutine Name:	LOC
2.	Purpose:	Compute a vector subscript for an element in a matrix of specified storage mode.
3.	Equations and Procedures:	NS=0 Subscript is computed for a matrix with N*M elements in storage (general matrix). NS=1 Subscript is computed for a matrix with N*(N+1)/2 in storage. (Upper triangle of symmetric matrix.) NS=2 Subscript is computed for a matrix with N element in storage (diagonal elements of a diagonal matrix.)
4.	Input Arguments:	I Row number of element J Column number of element M Number of runs in matrix N Number of columns in matrix NS One digit number for storage model of matrix 0 - General 1 - Symmetric 2 - Diagonal
5.	Output Arguments:	IR-Resultant vector subscript
6.	Error Returns:	None
7.	Calling Sequence:	Call LOC (I,J,IR,N,M,NS)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(492 bytes) 123 words
12.	Subroutine User:	MPRD,MSTR
13.	Subroutine Required:	None

1.	Subroutine Name:	MAXVAL
2.	Purpose:	General maximum absolute value of Eigenvector.
3.	Equations and Procedures:	The vector is searched to find maximum absolute component.
4.	Input Arguments:	DISP=Vector NON=Length of vector
5.	Output Arguments:	VAL=Actual algebraic value of maximum component
6.	Error Returns:	None
7.	Calling Sequence:	Call maxual (DISP,NDN,VAL)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(324 bytes) 81 words
12.	Subroutine User:	BASICD
13.	Subroutine Required:	None

1.	Subroutine Name:	MFGR
2.	Purpose:	
3.	Equations and Procedures:	Gaussian elimination technique is used for calculation of the triangular factors of a given matrix. Complete pivoting is built in.
4.	Input Arguments	A Given matrix M No. of rows of A N No. of cols. of A EPS Test value of zero affected by roundoff noise
5.	Output Arguments	A,IRANK,IROW,ICOL
6.	Error Returns:	None
7.	Calling Sequence:	Call MFGR(A,M,N,EPS,IRANK,IROW,ICOL)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(1974 bytes) 494 words
12.	Subroutine User:	AA
13.	Subroutine Required:	None

1.	Subroutine Name:	MFSD
2.	Purpose:	Factor a given symmetric positive definite matrix
3.	Equations and Procedures:	Solution obtained using the square root method of Cholesky. The given matrix is represented as product of two triangular matrices, where the left hand factor is the transpose of the right hand factor.
4.	Input Arguments:	A -Upper triangular part of the given symmetric positive definite N by N coefficient matrix. N -The number of rows (columns) in given matrix. EPS-An input constant which is used as relative tolerance for test on loss of significance.
5.	Output Arguments:	A -Contains resultant upper triangular matrix IER-Error return variable
6.	Error Returns:	If input parameter N is wrong or some radicand is non-positive then argument IER is set to -1 and return is made to calling routine.
7.	Calling Sequence:	Call MFSD (A,N,EPS,IER)
8.	Input Files:	None
9.	Output Files:	None
10.	Scratch Files:	None
11.	Storage Required:	(732 bytes) 183 words
12.	Subroutine User:	SINV
13.	Subroutine Required:	None

1.	Subroutine Name:	MINV
2.	Purpose:	Invert a matrix
3.	Equations and Procedures:	The standard Gauss-Jordan method is used. The determinant is also calculated. A determinant of zero indicates that the matrix is singular.
4.	Input Arguments:	A Matrix N Order of matrix L, M } Work storage each of length N
5.	Output Arguments:	A Resultant *inverse* D Resultant determinant
6.	Error Returns:	None
7.	Calling Sequence:	Call MINV (A,N,D,L,M)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(1874 bytes) 469 words
12.	Subroutine User:	S311D, S222, SUBA, S311
13.	Subroutine Required:	None

1.	Subroutine Name:	MINV2
2.	Purpose:	Control routine to invert a positive definite symmetric matrix.
3.	Equations and Procedures:	The upper triangle elements of a general matrix are used to form a symmetric matrix (subroutine MSTR). The matrix is inverted (subroutine SINV). The symmetric matrix is expanded to form a general matrix (subroutine MSTR).
4.	Input Arguments:	PHI Symmetric matrix to be inverted NX Number of rows (columns) in matrix W Work storage area
5.	Output Arguments:	PHI Resultant inverted matrix
6.	Error Returns:	None
7.	Calling Sequence:	Call MINV2 (PHI,NX,D,W,W1)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(642 bytes) 161 words
12.	Subroutine User:	S311
13.	Subroutine Required:	MSTR,SINV

1.	Subroutine Name:	MPRD
2.	Purpose:	Multiply two matrixes to form a resultant matrix.
3.	Equations and Procedures:	The M by L matrix B is premultiplied by the N by M matrix A and is stored in the N by L matrix R. This is a row into column product R is always output as a general matrix except when A and B are both diagonal matrices then R is output.
4.	Input Arguments:	A First input matrix B Second input matrix N No. of rows in A and R M No. of columns in A and rows in B MSA Storage mode of A 0-General 1-Symmetric 2-Diagonal MSB Same as MSA except for B L No. of columns in B and R
5.	Output Arguments:	R Output matrix cannot have same storage as A or B.
6.	Error Returns:	None
7.	Calling Sequence:	Call MPRD(A,B,R,N,M,MSA,MSB,L)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(944 bytes) 238 words
12.	Subroutine User:	SUBA
13.	Subroutine Required:	LOC

1.	Subroutine Name:	MRPRNT
2.	Purpose:	Matrix print routine
3.	Equations and Procedures:	Print an NR x NC matrix (row by row)
4.	Input Arguments:	NR number of rows NC number of columns AMTRIX an NR x NC matrix to be output to unit 101
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	Call MRPRNT (NR,NC,AMTRIX)
8.	Input Tapes:	None
9.	Output Tapes:	101 Unit 10
10.	Scratch Tapes:	None
11.	Storage Required:	(502 bytes) 126 words
12.	Subroutine User:	S461, EQSOL, S241, S311, TAPEII, S316, AA, OPTFR, S222, S316, S321, S451, S314
13.	Subroutine Required:	None

1.	Subroutine Name:	MSTR
2.	Purpose:	Change storage mode of a matrix
3.	Equations and Procedures:	The input matrix is restructured to form an output matrix.
4.	Input Arguments:	A Input matrix N Number of rows and columns MSA One digit code for storage mode of input matrix 0 - general, 1 - symmetric, 2 - diagonal MSR Same as MSA except for output matrix
5.	Output Arguments:	R Output matrix
6.	Error Returns:	None
7.	Calling Sequence:	Call MSTR (A,R,N,MSA,MSR)
8.	Input Files:	None
9.	Output Files:	None
10.	Scratch Files:	None
11.	Storage Required:	(542 bytes) 136 words
12.	Subroutine User:	MINV2
13.	Subroutine Required:	LOC

1. Subroutine Name: OPTFR

2. Purpose: Control program for optimization

3. Equations and Procedures:

a) Initialize variables and arrays.
b) If BASIC analysis call BASIC or BASICD and return.
c) Compute initial guess design - use input design or approximate fully stressed design from input design (call FSD, BASIC).
d) Initialize row and column vectors, and work array.
e) Compute row and column to store partials in work array (call assem).
f) Compute $\partial gA/\partial A$, $\partial gs/\partial A$, $\partial gs/\partial X$, $\partial gD/\partial A$, $\partial gD/\partial X$, $\partial gx/\partial A$, $\partial gx/\partial x$, $\partial gw/\partial A$ and store in work array (call S432, S435, S4310, S4323, S4325).
g) Initialize vectors for linear programming routine.
h) Determine which constraints will be active for optimum design and if linear design is feasible (call ITER).
i) Initialize row and column vectors for partial Newton-Raphson procedure.
j) Perform the partial Newton-Raphson procedure until design converges or maximum allowable iterations has occurred (cal S451).
k) If design has not converged repeat steps D-J-maximum of 3 iterations for linear programming phase.
kl) If dynamic constraints are to be included perform a basic analysis (call BASICD) and repeat steps D-K.
l) If design did not converge in linear programming phase compute a fully stressed design (call FSD) and initialize row and column vectors, check for displacement or frequency constraint violations, initialize ALAMBD.
m) If design converged in linear programming phase or design is fully stressed with no displacement or freq constraint violations then compute lambda for all "on" constraints (call EQSOL).
n) Verify that all values of lambda are ≥ 0.0. If not then set to 0.0 and turn constraint off.
o) If constraint violations or any lambda was <0.0 then perform the full Newton-Raphson procedure (call S461).
p) Print final results of optimization and return (call PRIN1, BASIC, BASICD).

4. Input Arguments: See glossary for definition.

5.	Output Arguments:	None
6.	Error Returns:	Writes - Program terminated due to singular matrix from SIMQ in OPTFR STOP
7.	Calling Sequence:	Call OPTFR(PRD,PRS,PRR,NTYPE,AMIN,WT,AREA,XC,YC, A,KL,P,NE,NDN,NX,NBOU,NSE,NODE,NDT,NDTNX,COND, ISTRES,NDC,NW,SIG,ND,IDYN,UP,LOW,IAREA,IMU, ALAMBD,W,WARAY,NSNL,NDCNL,NXNL,NWNL,NDNNL, NSENL,IRST,ICHECK,WDYN,PRI,MAXIT,WS,ARAY)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(16118 bytes) 4030 words
12.	Subroutine User:	LINK2
13.	Subroutine Required:	FSD,S432,PRINI,S435,EQSOL,S4310, ITER,S451,ASSEM,S4323,MRPRNT,REGEND, SIMQ,S461,BASIC,S4325,BASICD,SHIFTA,S4310B

1. Subroutine Name: PIVOT

2. Purpose: Perform a simplex pivot about a given matrix element.

3. Equations and Procedures:

   a) Interchange labels on row and column being pivoted.
   b) Update pivot element.
   c) Update pivot column excluding pivot element.
   d) Update general element excluding those in pivot row.
   e) Update pivot row excluding pivot element.
   f) Set matrix element to zero if less than the minimum tolerance.
   g) Pirnt intermediate Tableau.

4. Input Arguments:

A	Input matrix
IPR	Pivot row
IPC	Pivot column
LABR	Vector containing row labels
LABC	Vector containing column labels
MM	First declared dimension of "A" in calling program
M	Number of rows in matrix
N	Number of columns in matrix
TOL	Minimum absolute value allowed for element in matrix
IPFLAG	Intermediate printout code
IT	Previous number of iterations

5. Output Arguments:

A	Output matrix
LABR	Vector containing updated run labels
LABC	Vector containing updated column labels
IT	Updated iteration number

6. Error Returns: None

7. Calling Sequence: Call PIVOT(A,IPR,IPC,LABR,LABC,MM,M,N,TOL,IPFLAG,IT)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1110 bytes) 278 words

12. Subroutine User: ITER

13. Subroutine Required: INTPR

1.	Subroutine Name:	PRIN1
2.	Purpose:	Prints structure design variables and computes total weight.
3.	Equations and Procedures:	a. Print out design variables. b. Compute total structure weight and pr t.
4.	Input Arguments:	A,WT,NE See glossarv for definitions.
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	CALL PRIN1 (A,WT,NE)
8.	Input Tapes:	None
9.	Output File:	I02 - Unit 6 - Printer
10.	Scratch Tapes:	None
11.	Storage Required:	(638 bytes) 160 words
12.	Subroutine User:	OPTFR,S451,S461
13.	Subroutine Required:	None

1.	Subroutine Name:	REGEND
2.	Purpose:	Regenerate P_k (inertial load modes) and X_k (redundant modes) for recomputation of dynamic constraints.
3.	Equations and Procedures:	The mass matrix is computed first using the Z matrix and the A-vector. Then S324R is called to compute P_k in the W array. Return if NX = 0. If not, call S311D to assemble ϕ^{-1} and ψ. Finally call S314 to generate X_k.
4.	Input Arguments:	W - Contains DEL(NDN,NW) room for P_k(NDN,NW) and X_k(NX,NW) and last Z matrix (NE,NDN) as output from BASICD routine. NDN, NW, NX, NE, A - See Glossary. WI - Work storage
5.	Output Arguments:	P_k and X_k are inserted in the respective positions in the W array.
6.	Error Returns:	None
7.	Calling Sequence:	CALL RECEND (W,NDN,NW,NX,NE,A,WI)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(1048 bytes) 262 words
12.	Subroutine User:	OPTFR,S451,S461
13.	Subroutine Required:	S314 S311D S324R

1. Subroutine Name: SBTFD

2. Purpose: Compute $(B^1)^T * F * D1 = PSI\ \overline{\psi}$.

3. Equations and Procedures: If NX=0 return.

4. Input Arguments:
   - B1 - Element matrix from structure cutter matrix element.
   - F - Flexibility matrix
   - NP - No. of forces defined for a particular element.
   - W - Work storage (5 words max.)
   - D1 - Element matrix from structure cutter matrix.
   - NX,NDN - See Glossary

5. Output Arguments: PSI (NX,NDN) Resultant matrix (structural matrix ψ).

6. Error Returns: None

7. Calling Sequence: CALL SBTFD (B1,F,NP,NX,NDN,PSI,W,D1)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1026 bytes) 257 words

12. Subroutine User: S241

13. Subroutine Required: None

1. Subroutine Name: SB1IR

2. Purpose: Extract element and reaction B-matrices from structure cutter matrix.

3. Equations and Procedures: NS thru NF are used to extract rows from the A matrix which are located by the ICOL vector.

   $[b_1{}^i]$ is of order ($NP_i \chi Nx$) where NP_i is the no. of force components defined for the element.

   $[b_1{}^R]$ is of order (NBOU,NX) where NBOU is the no. of reactions for the problem.

4. Input Arguments:

NDT,NX,NDTNX	-	See Glossary
ICOL	-	Row decoding vector for a matrix.
A	-	Structure cutter matrix.
NS		Starting COL position of element or reactions in original A before structure cutting.
NF		Final COL position of element or reactions in original A before structure cutting.
NCOL		Length of ICOL

5. Output Arguments: W contains the $[b_1{}^i]$ or $[b_1{}^R]$ matrix by rows.

6. Error Returns: None

7. Calling Sequence: CALL SB1IR (ICOL,A,NDT,NX,NDTNX,NS,NF,NCOL,W)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (780 bytes) 195 words

12. Subroutine User: S241

13. Subroutine Required: None

1.	Subroutine Name:	SD1IR
2.	Purpose:	Extract element and reaction D-matrices from structure cutter matrix.
3.	Equations and Procedures:	NS thru NF are used to extract rows from the A matrix which are located by the ICOL vector. $[D_1{}^i]$ is of order (NP_i x NDN) where NP_i is the No. of force components defined for the element. $[D_1{}^i]$ is of order (NBOU X NDN) where NBOU is the No. of reactions for the problem.
4.	Input Arguments:	NDT,NDTNX,NDN,KL,NBOU-see glossary for definitions NS,NF-see procedure for definitions A-structure cutter matrix IROW,ICOL-logical vectors locating rows and cols. NCOL-length of ICOL
5.	Output Arguments:	W contains the $[D_1{}^i]$ or $[D_1{}^R]$ matrix by rows.
6.	Error Returns:	None
7.	Calling Sequence:	CALL SD1IR (IROW,ICOL,A,NDT,NDTNX,NS,NF,NDN,KL, NBOU,NCOL,W)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(938 bytes) 235 words
12.	Subroutine User:	S241
13.	Subroutine Required:	None

1.	Subroutine Name:	SHIFTA
2.	Purpose:	Shift elements in a general matrix and initialize elements in working matrix.
3.	Equations and Procedures:	W1(I)=W(I), W(I)=0.0
4.	Input Arguments:	W Matrix containing elements to be shifted to output matrix M Number of elements to shift MN Number of elements in matrix W to initialize
5.	Output Arguments:	W1 Output matrix
6.	Error Returns:	None
7.	Calling Sequence:	Call SHIFTA (W,W1,M,MN)
8.	Input Files:	None
9.	Output Files:	None
10.	Scratch Files:	None
11.	Storage Required:	(334 bytes) 84 words
12.	Subroutine User:	EQSOL, OPTFR, S461
13.	Subroutine Required:	None

1. Subroutine Name: SIMQ

2. Purpose: Obtain solution of a set of simultaneous linear equations.

3. Equations and Procedures: Method: Elimination using largest pivotal divisor.
   a) Forward solution
      1. Search for maximum coefficient in column (pivot).
      2. Test for pivot less than tolerance (singular matrix).
      3. Interchange rows if necessary.
      4. Divide equation by leading coefficient.
      5. Eliminate next variable.
   b) Perform Back solution.

4. Input Arguments: A - Matrix of coefficients stored columnwise size (N x N). Destroyed in computation.
   B - Vector of original constants (length = N).
   N - Number of equations and variables.

5. Output Arguments: B - Vector containing final solution values (length = N).

6. Error Returns: KS; 0 = normal solution, 1 = singular set of equations.

7. Calling Sequence: CALL SIMQ (A,B,N,KS)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1286 bytes) 322 words

12. Subroutine User: OPTFR, S451

13. Subroutine Required: None

1.	Subroutine Name:	SINV
2.	Purpose:	Invert a given symmetric positive definite matrix.
3.	Equations and Procedures:	Solution is obtained using the factorization by subroutine MFSD.
4.	Input Arguments:	A Upper triangular part of the given symmetric positive definite N by N coefficient matrix. N The number of runs and columns in matrix A EPS An input constant which is used as relative tolerance for test on loss of significance.
5.	Output Arguments:	A Resultant upper triangular matrix IER Error return variable
6.	Error Returns:	If input parameter N is wrong or some radicard is non-positive the argument IER is set to -1 and return is made to the calling routine.
7.	Calling Sequence:	Call SINV (A,N,EPS,IER)
8.	Input Files:	None
9.	Output Files:	None
10.	Scratch Files:	None
11.	Storage Required:	(826 bytes) 207 words
12.	Subroutine User:	MINV2
13.	Subroutine Required:	MFSD

1. Subroutine Name: SIMQ

2. Purpose: Obtain solution of a set of simultaneous linear equations.

3. Equations and Procedures: Method: Elimination using largest pivotal divisor.
   a) Forward solution
      1. Search for maximum coefficient in column (pivot).
      2. Test for pivot less than tolerance (singular matrix).
      3. Interchange rows if necessary.
      4. Divide equation by leading coefficient.
      5. Eliminate next variable.
   b) Perform Back solution.

4. Input Arguments:
   A - Matrix of coefficients stored columnwise size (N x N). Destroyed in computation.
   B - Vector of original constants (length = N).
   N - Number of equations and variables.

5. Output Arguments: B - Vector containing final solution values (length = N).

6. Error Returns: KS; 0 = normal solution, 1 = singular set of equations.

7. Calling Sequence: CALL SIMQ (A,B,N,KS)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1286 bytes) 322 words

12. Subroutine User: OPTFR, S451

13. Subroutine Required: None

1. Subroutine Name: STDB

2. Purpose: Compute $[B_1]^T [F][B_1] = \overline{\phi}$

3. Equations and Procedures: If NX=0 return.
Note B1 as input is actually $[B_1]^T$.

4. Input Arguments:
   - B1 - Element matrix from structure cutter matrix.
   - F - Element flexibility matrix.
   - NP - No. of force components for element.
   - NX - No. of redundants.
   - W - Working Storage.

5. Output Arguments: PHIB (NX,NX)

6. Error Returns: None

7. Calling Sequence: CALL STDB (B1,F,NP,NX,PHIB,W)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (962 bytes) 241 words

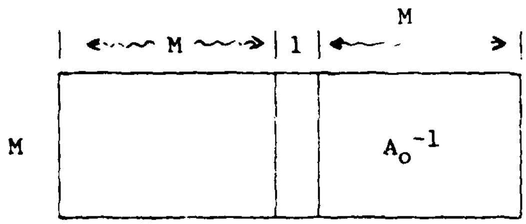
12. Subroutine User: S241

13. Subroutine Required: None

1. Subroutine Name: SUBA

2. Purpose: Generate $[A_o]^{-1}$ from structure cutter matrix.

3. Equations and Procedures: Extract lower triangular and upper triangular factor matrices and determine product.
   This product is A_o.
   Next get inverse of A_o.
   Store A_o^{-1} back into structure cutter matrix.



4. Input Arguments: A - Structure cutter matrix after routine MFGR.
   M - No. of rows in matrix A.
   N - No. of COLS in matrix A.
   WORK, WORK1 - Work storage

5. Output Arguments: A
   L - Lower triangle factor
   U - Upper triangle factor
   BO- A_o^{-1} x load

6. Error Returns: None

7. Calling Sequence: CALL SUBA (L,U,AO,WORK,WORK1,A,BO,M,N,WA)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1392 bytes) 448 words

12. Subroutine User: AA

13. Subroutine Required: MINV MPRD

1. Subroutine Name: S222

2. Purpose: Compute basic structural matrices for dynamic analysis.

3. Equations and Procedures: Read $\overline{\phi}$, $\overline{\psi}$ and $\overline{\Omega}$ from data set I13.

Use A and find

$$\phi = \sum_{1}^{NE} \frac{1}{A_i} \overline{\phi}_i \quad , \quad \psi = \sum_{1}^{NE} \frac{1}{A_i} \overline{\psi}_i \quad , \quad \Omega = \sum_{1}^{NE} \frac{1}{A_i} \overline{\Omega}_i$$

Get inverse of ϕ store on top of ϕ (MINV).

Form mass matrix from the Z matrix.

4. Input Arguments: A, NE, NX, NDN, - See Glossary for Definitions.
W - work storage.

5. Output Arguments: PHI, PSI, OMEGA - See Glossary for Definitions.
M - Mass matrix
Z - Z-matrix

6. Error Returns:

7. Calling Sequence: CALL S222 (PHI,PSI,OMEGA,W,A,NE,NX,NDN,M,Z)

8. Input Tapes: I13

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (2748 bytes) 687 words

12. Subroutine User: BASICD

13. Subroutine Required: MINV MRPRNT

1. Subroutine Name: S231

2. Purpose: Compute element forces or reactions as function of redundant X and load P.

3. Equations and Procedures:

$$\{S_i\} = [b_1{}^i][X] + [D_i][P] \quad (2.3.1)$$

$$\{R\} \quad [b_1{}^R][X] + [D^R][P] \quad (2.3.1)$$

4. Input Arguments: NTYPE,NE,NDT,NDTNX,X,NX,NDN,P,NBOU,KL,NCOL,NLL,NSE - See Glossary for Definitions.

NW - Length of S output.

5. Output Arguments: S - Either forces or reactions (NSR > 0 or NSR < 0 respectively).

6. Error Returns: None

7. Calling Sequence: CALL S231 (NTYPE,I12,NE,NDT ,NDTNX,X,NX,NDN, P,NSR,NBOU,KL,W,NW,NCOL,S,NLL,NSE)

8. Input Tapes: I12, I15

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (2012 bytes) 503 words

12. Subroutine User: BASICD, BASIC

13. Subroutine Required: S231A

1.	Subroutine Name:	S231A
2.	Purpose:	Computes start and final columns of element B matrix in a (structure cutter matrix).
3.	Equations and Procedures:	Take into account the following facts about the structure cutter matrix. 1) The reactions are in the first NBOU columns. 2) Axial and shear web elements have only 1 force column in B-element matrix. 3) Triangle has 3 columns and quadrilateral element has 5 columns in their B element matrices.
4.	Input Arguments	NTYPE Element type (1-4) NE No. of elements NSR Element No. NBOU No. of reactions
5.	Output Arguments:	NS,NF starting and final column no's. respectively.
6.	Error Returns:	None
7.	Calling Sequence:	Call S231A (NTYPE,NE,NSR,NS,NF,NBOU)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(574 bytes) 144 words
12.	Subroutine User:	S241,S231
13.	Subroutine Required:	None

1. Subroutine Name: S241

2. Purpose: Compute element normalized matrices and put on tape. Write b_1 and D_1 matrices to tape.

3. Equations and Procedures:

$$[\bar{\phi}i] = [b_1{}^i]^T [\bar{f}_i] [b_1{}^i] \quad (NX \times NX) \qquad (2.4.1)$$

$$[\bar{\psi}i] = [b_1{}^i]^T [\bar{f}_i] [D_i] \quad (NX \times N_{DN}) \qquad (2.4.2)$$

$$[\bar{\Omega}_i] = [D_i]^T [\bar{f}_i] [D_i] \quad (N_{DN} \times N_{DN}) \qquad (2.4.3)$$

BMG21 is used to compute the Z matrix a row of which is put on each bar matrix record of unit I13.

4. Input Arguments: NTYPE,KL,NE,NBOU,NDT,NDTNX,NDN,NX,NCOL-see glossary

IROW	Row label vector (structure cutter)
ICOL	Column label vector (structure cutter)
A	A matrix (structure cutter)
NTAPE	Output file unit number
W	Work array
BMG21	Element mass contribution vector

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence: Call S241,(NTYPE,IROW,ICOL,A,KL,NE,NBOU,NDT, NDTNX,NDN,NX,NTAPE,NCOL,W,BMG21)

8. Input Tapes: NSS1

9. Output Tapes: I13, I15,I12

10. Scratch Tapes: None

11. Storage Required: (3228 bytes) 807 words

12. Subroutine User: TAPE11

13. Subroutine Required: STDB,SBTFD,SB1IR,SD1IR, S231A,MRPRNT,TRPRNT

1. Subroutine Name: S311

2. Purpose: Assembles bar matrices into ϕ, ψ and Ω.

3. Equations and Procedures:

$$[\phi] = \sum_{i=1}^{NE} \frac{1}{Ai} [\bar{\phi}_i] \qquad (3.1.1)$$

$$[\psi] = \sum_{i=1}^{NE} \frac{1}{Ai} [\bar{\psi}_i] \qquad (3.1.2)$$

$$[\Omega] = \sum_{i=1}^{NE} \frac{1}{Ai} [\bar{\Omega}_i] \qquad (3.1.3)$$

4. Input Arguments: A,NE,NX,NDN - see glossary
W- work storage

5. Output Arguments: PHI,PSI,OMEGA - see glossary

6. Error Returns: None

7. Calling Sequence: Call S311(PHI,PSI,OMEGA,W,A,NE,NX,NDN)

8. Input Tapes: I13

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (2146 bytes) 537 words

12. Subroutine User: BASIC

13. Subroutine Required: MINV2 ,MRPRNT

1. Subroutine Name: S311D

2. Purpose: Compute PHI^{-1} and PSI using (3.1.1) and (3.1.2) for dynamics analysis.

3. Equations and Procedures:

$$[\phi] = \sum_{i=1}^{NE} \frac{1}{Ai} [\bar{\phi}_i] \qquad (NX \times NX)$$

$$[\psi] = \sum_{i=1}^{NE} \frac{1}{Ai} [\bar{\psi}_i] \qquad (NX \times N_{DN})$$

4. Input Arguments: A,NE,NX,NDN - see glossary
W -work storage

5. Output Arguments: PHI,PSI - see glossary

6. Error Returns: None

7. Calling Sequences: Call S311D (PHI,PSI,W,A,NE,NX,NDN)

8. Input Tapes: I13

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1298 bytes) 325 words

12. Subroutine User: REGEND

13. Subroutine Required: MINV

1.	Subroutine Name:	S314
2.	Purpose:	Computes redundants
3.	Equations and Procedures:	$\{X\} = - [\phi]^{-1} [\psi] \{P\}$ (3.1.4)
4.	Input Arguments:	P,NX,NDN,NLL,PHI,PSI,NLL-see glossary W-work storage
5.	Output Arguments:	X Vector containing computed redundants
6.	Error Returns:	None
7.	Calling Sequence:	Call S314 (PHI,PSI,P,X,NX,NDN,W,NLL)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(1164 bytes) 291 words
12.	Subroutine User:	REGEND,BASICD,BASIC
13.	Subroutine Required:	MRPRNT

1.	Subroutine Name:	S315
2.	Purpose:	Compute reduced displacement vector - static analysis.
3.	Equations and Procedures:	$\{\Delta\} = [\psi]^T \{X\} + [\Omega] \{P\}$ (3.1.5)
4.	Input Arguments:	NLL,PSI,OMEGA,X,P,NDN,NX,NLL-see glossary
5.	Output Arguments:	DISP Vector containing displacements
6.	Error Returns:	None
7.	Calling Sequence:	Call S315 (PSI,X,OMEG,A,P,DISP,NDN,NX,NLL)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(882 bytes) 221 words
12.	Subroutine User:	BASIC
13.	Subroutine Required:	None

1.	Subroutine Name:	S316
2.	Purpose:	Calculate flexibility matrix for dynamics analysis.
3.	Equations and Procedures:	$[F] = [\Omega] - [\psi]^T [\phi^{-1}] [\psi]$ (3.1.6) (NDN x NDN)
4.	Input Arguments:	W Work storage OMEGA,PSI,PHI,NX,NDN - see glossary
5.	Output Arguments:	OMEGA Replaced by {F}
6.	Error Returns:	None
7.	Calling Sequence:	Call S316 (OMEGA,PSI,PHI,NX,NDN,W)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(1032 bytes) 256 words
12.	Subroutine User:	BASICD
13.	Subroutine Required:	MRPRNT

1.	Subroutine Name:	S321
2.	Purpose:	Calculate Eigenvalue matrix for a dynamic analysis.
3.	Equations and Procedures:	$Q_{ij} = F_{ij} (M_i M_j)^{\frac{1}{2}}$ (3.2.1)
4.	Input Arguments:	F Flexibility matrix M Mass matrix (only diagonal defined) NDN Reduced DOF
5.	Output Arguments:	F Replaced by Q
6.	Error Returns:	None
7.	Calling Sequence:	Call S321 (F,M,NDN)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(680 bytes) 170 words
12.	Subroutine User:	BASICD
13.	Subroutine Required:	MRPRNT

1.	Subroutine Name:	S323
2.	Purpose:	Calculate inertia load mode for dynamic analysis.
3.	Equations and Procedures:	$[P_k{}^i] = N_i{}^{1/2}\, q_k{}^i$ (3.2.1) $W = {}^1/\lambda$ (3.2.2)
4.	Input Arguments:	M — Mass matrix Q — q (Eigen vector) NDN — Red DOF NW — No of modes Lamda — λ
5.	Output Arguments:	Q - replaced by P_K
6.	Error Return:	None
7.	Calling Sequence:	Call S323 (M,Q,NDN,NW,LAMDA)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(540 bytes) 135 words
12.	Subroutine User:	BASICD
13.	Subroutine Required:	None

1.	Subroutine Name:	S324
2.	Purpose:	Calculate displacement modes in dynamic analysis.
3.	Equations and Procedures:	$\Delta_K^{\ i} = P_K^{\ i} / w_K^2 \ M_i$ (3.2.4)
4.	Input Arguments:	P Inertial load mode M Mass matrix (diagonal) NDN Red DOF NW No of modes W Work storage
5.	Output Arguments:	DEL - Displacement modes
6.	Error Returns:	None
7.	Calling Sequence:	Call S324 (F,M,NDN,DEL,NW,W)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(528 bytes) 132 words
12.	Subroutine User:	BASICD
13.	Subroutine Required:	None

1. Subroutine Name: S324R

2. Purpose: Solves for P_K from Δ_K for dynamics reanalysis (optimization).

3. Equations and Procedures: $P_K^i = \Delta_K^i * M_i$

4. Input Arguments:

DEL	Displacement modes
M	Mass matrix
NW	No. of modes
NDN	No. of reduced DOF

5. Output Arguments: PK Inertial lode mode.

6. Error Returns: None

7. Calling Sequence: Call 5324R (DEL,M,PK,NW,NDN)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (470 bytes) 118 words

12. Subroutine User: REGEND

13. Subroutine Required: None

1. Subroutine Name: S4310

2. Purpose: Computes 1st partial of $g\sigma$ with respect to X.

3. Equations and Procedures: Each row of the NE x N_X matrix is

$$\underset{N_{pi}}{\frac{1}{A_i \sigma_i^*} \frac{\partial Yi}{\partial Si}} \quad \underset{N_{pi} x N_X}{[b_1^i]} \qquad (4.3.10)$$

4. Input Arguments: A,SIG,Y,S,NTYPE,AREA,XC,YC,NE,NSE,NDT,NX,IR,IC, NTRANS,NR,NC,NCOL,NDTNX,NROW,NBOU - see glossary

   W work storage

5. Output Arguments: Each element computed is inserted in ARAY by S4310A.

6. Error Returns: None

7. Calling Sequence: Call S4310 (A,SIG,Y,S,NTYPE,AREA,XC,YC,NE, NSE,NDT,NX,IR,IC,NTRANS,ARAY,NR,NC,NCOL, NDTNX,NROW,NBOU,W)

8. Input Tapes: I12

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (2600 bytes) 650 words

12. Subroutine User: OPTFR,S451,S461,EQSOL

13. Subroutine Required: SQRT,S4310A,TRPRNT

1.	Subroutine Name:	S4310A
2.	Purpose:	Output of calculated matrix element to a temporary file.
3.	Equaticns and Procedures:	Write row number, column number, transpose control, element value and summing control to a temporary data file.
4.	Input Arguments:	IB1 Row number IB2 Column number NT Transpose/symmetry control ANS Element data NS Summary control
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	Call S4310A(A,NR,NC,IB1,IB2,NT,ANS,NS)
8.	Input Files:	None
9.	Output Files:	I14
10.	Scratch Files:	None
11.	Storage Required:	(378 bytes) 95 words
12.	Subroutine User:	S4614,S4621,S4622,S432,S435,S4323,S4310, S4325,S466
13.	Subroutine Required:	None

1. Subroutine Name: S4310B

2. Purpose: Insert calculated element into analysis matrix

3. Equations and Procedures: Reads element (ANS), row and column positions (IB1,IB2), transpose/lower triangle control (NT), and summing control (NS) from input file. Stores elements in output matrix "A" as specified by control variables

NT	NS=0	=1
+1	A(IB1,IB2)=ANS	A(IB1,IB2)=A(IB1,IB2)+ANS
+2	A(IB2,IB1)=ANS	A(IB2,IB1)=A(IB2,IB1)+ANS
-1	A(IB1,IB2)=ANS (for all IB1≤IB2)	A(IB1,IB2)=A(IB1,IB2)+ANS
-2	A(IB2,IB1)=ANS (for all IB1≤IB2)	A(IB2,IB1)=A(IB2,IB1)+ANS

4. Input Arguments:
   - A Matrix to be assembled
   - M Number of rows
   - N Number of columns

5. Output Arguments: A Assembled matrix

6. Error Returns: None

7. Calling Sequence: Call S4310B (A,M,N)

8. Input Files: 114

9. Output Files: None

10. Scratch Files: None

11. Storage Required: (798 bytes) 200 words

12. Subroutine User: EQSOL,OPTFR,S451,S461

13. Subroutine Required: None

1. Subroutine Name: S432

2. Purpose: Generates 1st partial of g_A with respect to A. Also G_A is computed if requested.

3. Equations and Procedures:

$$g_A^{\,i} = \frac{Ai^*}{Ai} - 1 \quad \text{IF IGA} \neq 0 \qquad (4.3.1)$$

$$\frac{\partial g_A^{\,i}}{\partial A_j} = \frac{Ai^*}{A_j^{\,2}}\,\delta ij \quad \text{(diagonal matrix)} \qquad (4.3.2)$$

4. Input Arguments: AMIN,A,NE,IR,IC,NTRANS,NR,NC,IGA - see glossary

5. Output Arguments: ARAY matrix for storage of computed partial
VECT Vector for storage of g_A

6. Error Returns: None

7. Calling Sequence: Call S432 (AMIN,A,NE,IR,IC,NTRANS,ARAY,NR,NC, VECT,IGA)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (794 bytes) 199 words

12. Subroutine User: OPTFR,EQSOL

13. Subroutine Required: S4310A

1. Subroutine Name: S4323

2. Purpose: Generates 4 partials $\frac{\partial g}{\partial A}$, $\frac{\partial g\Delta}{\partial x}$, $\frac{\partial gx}{\partial A}$, $\frac{\partial gx}{\partial x}$ and $g\Delta$ and gx

3. Equations and Procedures:

$$g\Delta = \frac{1}{\Delta^*} (\{\bar{\Psi}_i\}^T \{x\} + [\bar{\Omega}^i] \{P\}) - 1 \qquad (4.3.21)$$

$$gx = [\phi] \{x\} + [\psi] \{P\} \qquad (4.3.35)$$

$$\frac{\partial g^i}{\partial A_j} = - \frac{1}{\Delta^*_i A_j 2} [\ddot{\bar{\psi}}(j)_i X + \tilde{\bar{\Omega}}(j) mi Pi] \qquad (4.3.23)$$

$$\frac{\partial g\Delta^i}{\partial xk} = \frac{1}{\Delta^*_i} \bar{\Psi} ki \qquad (4.3.23)$$

$$\frac{\partial gx}{\partial A} = - \frac{1}{Aj^2} ([\bar{\Phi}_j] \{x\} + [\bar{\Psi}_j] \{P\}) \text{5th col.} \qquad (4.3.38)$$

$$\frac{\partial gx^i}{\partial xk} = \phi_{ik} \qquad (4.3.40)$$

4. Input Arguments: A,NE,P,NDN,NDC,NX,NR,NC,ND,X,NTRANS-see glossary

IR1	Vector containing displacement constraint row labels
IR2	Vector containing redundant constraint row labels
IC1	Vector containing unknown design variable column labels
IC2	Vector containing redundant variable column labels
DSTARU	Vector containing upper displacement limits
DSTARL	Vector containing lower displacement limits
W	Work storage
IGX	Control to compute gx
IGD	Control to compute gΔ

5. Output Arguments: ARAY is modified by the partials
VECT Storage for gΔ and gx

6. Error Returns: None

7. Calling Sequence: Call S4323 (A,NE,P,NDN,NDC,NX,IR1,IR2,IC1,IC2, ARAY,NR,NC,DSTARU,DSTARL,ND,X,W,NTRANS,VECT, IGX,IGD

8. Input Tapes: I13

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (3483 bytes) 871 words

12. Subroutine User: OPTFR,S451,S461,EQSQL

13. Subroutine Required: S4310A

1. Subroutine Name: S4325

2. Purpose: Generate Frequency Constraint Optimizing Factors,

$$\frac{\partial g_w}{\partial A}, \quad g_w, \quad \frac{\partial g_w^2}{\partial A, A_K} \quad \text{summed into} \quad \frac{\partial L^2}{\partial A_j \partial A_K}$$

3. Equations and Procedures:

$$g_w^i = \frac{1}{A_t} \bar{\Phi}^t_{\ell_m} (X_K)_\ell (X_K)_m + \frac{2}{A_t} \bar{\Psi}^t_{np} (X_K)_n (P_K)_p + \frac{1}{A_t} \bar{\Omega}^t_{qr} (P_K)_q (P_K)_r$$

$$- \frac{1}{(w_i^*)^2} \frac{1}{Ag\ \bar{m}g\ \beta_{gs}} (P_K)_s^2 \qquad (4.3.31)$$

$$\frac{\partial g_w{}^i}{\partial A_j} = \frac{1}{A_j{}^2} [\{X_K\}^T [\bar{\Phi}_j] \{X_K\} + 2 \{X_K\}^T [\bar{\Psi}_j] \{P_K\} + \{P_K\}^T [\bar{\Omega}_j] \{P_K\}]$$

$$+ \frac{1}{(\omega_i)^2} \{P_K\}^T [Z_j] \{P_K\} \quad \text{Where} \quad Z = \underset{NE}{\overset{non}{[Z]}} \frac{\beta_{jsmj}}{M_s{}^2} \qquad (4.3.32)$$

$$\frac{\partial^2 g_w{}^i}{\partial A_j \partial A_K} = \frac{2}{A_j{}^3} [\bar{\Phi}^j_{\ell_m} (X_i)_\ell (X_i)_m + 2\bar{\Psi}^j_\ell (X_i)_n (P_i)_p + \bar{\Omega}^j_{qr} (P_i)_q (P_i)_r] \delta_{jK}$$

$$- \frac{2}{(w_i^*)^2} \frac{\beta_{js\bar{m}j} \beta_{Ks\bar{m}K}}{(\beta_{gs\bar{m}g}\ Ag)^3} (\rho_i)_S^2$$

4. Input Arguments:

A,WS,NE,NDN,NX,NR,NC - See Glossary for definitions
PK - Vector containing inertial load mode
XK - " " redundant mode
NWC - Number of modes
W - Work storage
Z - Matrix containing mass contributions of elements/DOF
IMU - Vector identifying active freq. constraints

MUW - Vector containing frequency μ's
IRI - Assembly vector containing row labels $(\partial gw^2/\partial A\partial A)$
IC2 - Assembly vector containing column labels $(\partial gw^2/\partial A\partial A)$, $\partial gw/\partial A$
IGP - Assembly vector containing row labels $(\partial w/\partial A)$
NGW - Control switch to compute GW
NDW - " " " " $\partial GW^2/\partial A\partial A$
NT - Transpose control

5. Output Arguments: ARAY - External matrix used for linear programming and Newton Raphson solutions GW - vector to store computed gw values

6. Error Returns: None

7. Calling Sequence: Call S4325 (A,WS,PK,XK,NWC,NE,NDN,NX,W,GW, ARAY,NR,NC,Z,IMU,MUW,IR1,IC2,IGP,NGW,NDW,NT)

8. Input Tapes: I13

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (3228 BYTES) 807 words

12. Subroutine User: OPTFR,S451,S461,EQSOL

13. Subroutine Required: S4310A,S4325A,MRPRNT

1. Subroutine Name: S4325A

2. Purpose: Compute f_i for g_w and both partials of g_w

3. Equations and Procedures:

$$f_i = \{X_K\}^T [PHI] \{X_K\} + 2.0 \{X_K\}^T [PSI] \{P_K\}$$

$$+ \{P_K\}^T [OMEGA] \{P_K\} \qquad (4.3.26)$$

4. Input Arguments: PHI,PSI,OMEGA,NX,NON - See Glossary for definition
X - Vector containing redundant mode
P - Vector containing inertial load mode
W - work storage

5. Output Arguments: ANS - f_i is returned to calling program in this variable

6. Error Returns:

7. Calling Sequence: Call S4325A(PHI,PSI,OMEGA,X,P,ANS,NX,N N,W)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: 1096 bytes (274 words)

12. Subroutine User: S4325

13. Subroutine Required: None

1. Subroutine Name: S433

2. Purpose: Compute stresses and Ψ_i for an element and print stresses if requested.

3. Equations and Procedures: Stresses are computed as follows,

Bar element $S_x = \frac{S_1}{A}$ $S_y = 0$ $T_{xy} = 0$

Shear Panel $S_x = 0$ $S_y = 0$ $T_{xy} = \frac{S_1}{A\sqrt{AREA}}$

Triangle $S_x = \frac{S_1}{A\sqrt{AREA}}$, $S_y = \frac{S_2}{A\sqrt{AREA}}$, $T_{xy} = \frac{S_3}{A\sqrt{AREA}}$

Quad $S_x = \frac{S_1}{A\sqrt{AREA}} + \frac{S_2\, y_c}{A*AREA}$, $S_y = \frac{S_3}{A\sqrt{AREA}} + \frac{S_4\, X_c}{A*AREA}$

$$T_{xy} = \frac{S_5}{A\sqrt{AREA}}$$

$$Y_i = A_i \sqrt{S_x^2 + S_y^2 - S_x S_y + 3 * T_{xy}^2}$$

4. Input Arguments:
   - PRS - Print control
   - NTYPE - Element type
   - S - Element forces
   - A - Element design parameter
   - AREA - Surface area of element
   - X_c, Y_c - Coordinates of midpoint of quad
   - I - Element number
   - II - Load condition number

5. Output Arguments: Y_I - Mises - Hencky criteria for the element

6. Error Returns: None

7. Calling Sequence: Call S433(PRS,NTYPE,S,A,AREA,X_C,Y_C,Y_I, I,II,)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (1344 BYTES) 336 words

12. Subroutine User: BASIC, BASICD

13. Subroutine Required: None

1. Subroutine Name: S435

2. Purpose: Generates partial of g_σ with respect to A and g_σ

3. Equations and Procedures:

$$g_c^i = \frac{Y_i}{A_j \sigma_i^*} - 1 \quad ; \quad \frac{\partial g\sigma^i}{\partial A_j} = - \frac{Y_i}{A_j^2 \sigma_i *} \delta_{ij}$$

a) For each unknown design and active stress constraint compute $\partial g_\sigma{}^i/\partial A_j$. Assemble in coefficient matrix using routine S4310A.

b) For each active stress constraint compute g_σ and assemble in constant vector.

4. Input Arguments: A,SIG,Y,NE,IR,IC,NTRANS,NR,NC,IGS
See Glossary for definitions.

5. Output Arguments: ARAY - Assembled matrix containing $\partial G_s/\partial A$
VECT - Vector containing g_s values

6. Error Returns: None

7. Calling Sequence: Call S435(A,SIG,Y,NE,IR,IC,NTRANS,ARAY,NR,NC,VECT,IGS)

8. Inpur Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (842 BYTES) 211 words

12. Subroutine User: OPTFR, S451, S461, EQSΦL

13. Subroutine Required: S4310A

1. Subroutine Name: S451

2. Purpose: Perform a partial Newton Raphson

3. Equations and Procedures:
   a. Compute matrix size
   b. Print design variables and perform a basic analysis and update dynamic variables. Call PRINI,BASIC,REGEND
   c. Iteration loop
      1) Initilize working arrays
      2) Compute and assemble the partials G*S/A+, G*S/X, G*D/A+, G*D/X, GX/A+, GX/X, G*W/A+ into array WARAY and the constraints G*S, G*D, GX and G*W into VECTOR "G" (Call ASSEM, S435, S4310, S4323, S4325)
      3) Print intermediate data if requested
      4) Solve for Delta A and Delta X. Call SIMQ
      5) Check for convergence of all A and X and update using New = Old + Delta
      6) Print updated design (call PRINI)
      7) Perform a static analysis and update dynamic variables (Call BASIC,REGEND)
      8) If any variable did not converge and design variables did not fall below lower limits and less than MAXIT iterations have been performed then repeat loop.
   d. Check all constraints for violations. If violations occur set error switch, raise design variables, print new design and perform a static analysis and update dynamic variables (call PRINI,BASIC,REGEND)

4. Input Arguments: PRD,PRS,PRR,NASTAR,NSSTAR,NDSTAR,NXSTAR,NWSTAR,W, G,WARAY,IYI,A,SIG,NTYPE,AREA,XC,YC,NE,NSE,NDT,NX, NCOL,NOTNX,NBOU,ISI,NDC,IMUSL,IMUDL,IAREA,IMU, P,NDN,UP,LOW,IX,IDYN,NW,IMUXL,IMUWL,ICC,ALAMBD, AMIN,WT,KL,NODE,ND,COND,IDEL.NSNL,NDCNL,NXNL, NWNL,ICHECK,WDYN,PRI,WS,MAXIT,IPK,IXK,IZ
   See Glossary for definition.

5. Output Arguments: W,A,WDYN, See Glossary for definition

6. Error Returns: Write: MATRIX is singular IN S451 - Partial N-R... STOP IERR: 0 = converged, 1 = did not converge, 2 = constraint violation

7. Calling Sequence: Call S451(PRD,PRS,PRR,NASTAR,NSSTAR,NDSTAR,NXSTAR, NWSTAR,W,G,WARAY,IYI,A,SIG,NTYPE,AREA,XC,YC,NE, NSE,NDT,NX,NCOL,NDTNX,NBOU,ISI,NDC,IMUSL,IMUDL, IAREA,IMU,P,NDN,UP,LOW,IX,IDYN,NW,IMUXL,IMUWL, IERR,ICC,ALAMBD,AMIN,WT,KL,NODE,ND,COND,IDEL,NSNL, NDCNL,NXNL,NWNL,ICHECK,WDYN,PRI,WS,MAXIT,IPK,IXK, IZ,ARAY)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (7246 BYTES) 1812 words

12. Subroutine User: OPTFR

13. Subroutine Required: SIMQ,S435,ASSEM, BASIC,PRINI,S4310,S4323, S4325, TESTR,MRPRNT,REGEND,S4310B

1. **Subroutine Name:** S461

2. **Purpose:** Performs a full Newton-Raphson.

3. **Equations and Procedures:**

a) Iteration loop
  1) Compute matrix dimensions and initialize working storage [illegible]
  2) Compute and assemble the partials G*S/A+, G*S/X,G*D/A+,G*[illegible], [illegible]X/A+,GX/X,G*W/A+,G2*W/A+A+, and the constraints [illegible]*D,GX,G*W and store in matrix WARAY.
     (CALL ASSEM,S435,S4310,S4323,S4325)
  3) Compute and assemble the partials G2L/A+A+, G2L/A+X,G2L/XX and store in matrix WARAY.
     (CALL ASSEM,S4614,S4621,S4622)
  4) Compute and assemble the partials GL/A+ and GL/X and store in matrix WARAY.
  5) Print lower triangle matrix WARAY if requested.
  6) Solve for DELTA A+, DELTA X and DELTA U* using routine GELS.
  7) Check for convergence of all design variables and redundants and update to new values. Set error switch if any did not converge.
  8) Check for convergence of all U*, update to new values, verify all U* are positive, if not positive set to 0.0 and turn constraint OFF.
  9) Verify that all design variables are greater than 0.0. If not set to minimum and turn constraint "ON".
  10) Print new design, perform a static analysis and update dynamic variables. CALL PRIN1,BASIC REGEND.
  11) If any variable did not converge and the number of iterations is less than maximum, repeat iteration loop.

b) If design converged check all constraints:
  1) Check all displacement constraints for violations and turn "ON" if violated.
  2) Check all stress constraints for violations and turn "ON" if violated.
  3) If any violations occurred, repeat Step A.

c) Compute the LaGrange multipliers for all design constraints which are "ON" using routines ASSEM, S435,S4323,S4325, and S432.

d) Compute total weight from LaGrange multipliers.

4. **Input Arguments:** PRD,PRS,PRR,NASTAR,NSSTAR,NDSTAR,NXSTAR,NWSTAR,NE,NX, W,WARAY,IAREA,IMU,IMUAL,IMUSL,IMUDL,IMUXL,IMUWL,A,SIG, IX,IYI,ISI,IDEL,NTYPE,AREA,XC,YC,NSE,NDT,NCOL,NDTNX, NBOU,NDC,P,NDN,UP,LOW,ND,NW,ALAMBD,WT,COND,AMIN,KL, NODE,ICC,NSNL,NDCNL,NXNL,NWNL,WDYN,PRI,WS,MAXIT.IPK, IXK,IZ,IDYN
See Glossary for Description.

5. **Output Arguments:** W,A,WDYN - See Glossary for Definitions.

6. Error Returns: Write: Matrix is singular in S461 - full N.r-...
IERR STOP

7. Calling Sequence: CALL S461 (PRD, PRS, PRR,NASTAR,NSSTAR,NDSTAR,NXSTAR, NWSTAR,NE,NX,W,WARAY,IAREA,IMU,IMUAL,IMUSL,IMUDL, IMUXL,IMUWL,A,SIG,IX,IYI,ISI,IDEL,NTYPE,AREA,XC,YC, NSE,NDT,NCOL,NDTNX,NBOU,NDC,P,NDN,UP,LOW,ND,NW,ALAMBD, WT,IERR,COND,AMIN,KL,NODE,ICC,NSNL,NDCNL,NXNL,NWNL, ICHECK,WDYN,PRI,WS,MAXIT,IPK,IXK,IZ,IDYN,ARAY)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: (14040 bytes) 3510 words

12. Subroutine User: OPTFR

13. Subroutine Required: GELS,S435,ASSEM,BASIC,PRIN1,S4310,S4323,S4325, S4614,S4621,S4622,TESTR,MRPRNT,REGEND,SHIFTA,S4310B

1. Subroutine Name: S4614

2. Purpose: Second partials with respect to $A_J A_K$.

3. Equations and Procedures:

$$\frac{\partial^2 g_\sigma^i}{\partial A_J \partial A_K} = \frac{2 Y_i}{A_i^3 \sigma_i^*} \delta_{iJ} \delta_{iK} \qquad (4.6.4)$$

$$\frac{\partial^2 g_\Delta^i}{\partial A_J \partial A_K} = \frac{2}{A_J^3 \Delta_i^*} [\bar{\psi}_{(J)\ell i} X_\ell + \bar{\bar{\Omega}}_{(J)mi} P_m] \delta_{JK} \qquad (4.6.8)$$

$$\frac{\partial^2 g_X^i}{\partial A_J \partial A_K} = \frac{2}{A_J^3} [\bar{\Phi}_{i\ell}^J X_\ell + \ldots_{im} P_m] \delta_{JK} \qquad (4.6.12)$$

$$\frac{\partial^2 L}{\partial A_J^+ \partial A_K^+} = \mu_\sigma^{i*} \frac{\partial^2 g_\sigma^{i*}}{\partial A_J^+ \partial A_K^+} + \mu_\Delta^{i*} \frac{\partial^2 g_\Delta^{i*}}{\partial A_J^+ \partial A_K^+} + \mu_X^i \frac{\partial^2 g_X^i}{\partial A_J^+ \partial A_K^+}$$

$$+ \mu_W^{i*} \frac{\partial^2 g_W^{i*}}{\partial A_J^+ \partial A_K^+} \qquad (4.6.20)$$

NOTE: The 4th term in Eq. 4.6.20 is summed into the coefficient matrix by routine S4325.

a) Each term in Eq. 4.6.20 is computed and summed into the lower triangular coefficient matrix by routine S4310A.

4. Input Arguments: NR,NC,A,SIG,X,P,UP,LOW,ND,NE,NDC,NX,NDN,W

MUS = Vector containing μ_σ
MUD = Vector containing μ_Δ
MUX = Vector containing μ_X
YI = Element force vector
IRA = Output ROW and COL vector
IRS = Stress constraint active vector
IRD = Displ. constraint active vector
IRX = Redundant constraint active vector
W = Working storage vector

5. Output Arguments: ARAY - Coefficient matrix

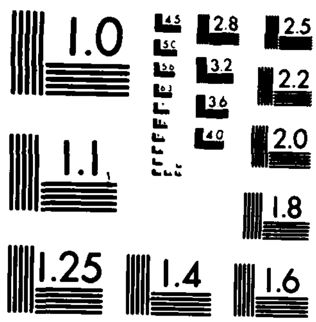
6. Error Returns: None

7. Calling Sequence: CALL S4614 (ARAY,NR,NC,MUS,MUD,MUX,YI,A,SIG,X,P,UP,LOW,ND,NE,NDC,NX,NDN,IRA,IRS,IRD,IRX,W)

AD-A128 235 FORCE METHOD OPTIMIZATION II VOLUME II USER'S MANUAL (U) BELL AEROSPACE TEXTRON BUFFALO NY J R BATT ET AL. NOV 82 AFWAL-TR-82-3088-VOL-2 F33615-8-C-3214 3/3

UNCLASSIFIED F/G 9/2 NL

END
DATE
FILMED
DTIC


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

8. Input Tapes: Unit 13 containing $\overline{\Phi}$ $\overline{\Psi}$ & $\overline{\Omega}$.

9. Output Tapes: None

10: Scratch Tapes: None

11. Storage Required: (2338 bytes) 577 words

12. Subroutine User: S461

13. Subroutine Required: S4310A

1. **Subroutine Name:** S4621

2. **Purpose:** Compute $\frac{\partial L^2}{\partial A_J \partial X_K} = \mu_\sigma^{i*} \frac{\partial^2 g_\sigma^{i*}}{\partial A_J^+ \partial X_K} + \mu_\Delta^{i*} \frac{\partial^2 g_\Delta^{i*}}{\partial A_j^+ \partial X_K} + \mu_x^i \frac{\partial^2 g_x^i}{\partial A_j^+ \partial X_K}$

3. **Equations and Procedures:** Each partial in the expression is computed and summed into the lower symmetric analysis matrix ARAY

$$\frac{\partial^2 g_\sigma^i}{\partial A_J \partial X_K} = -\frac{1}{A_i{}^2 \sigma_i{}^*} \frac{\partial Yi}{\partial S_i{}^{(\ell)}} b_1{}^i{}_{\ell K} \delta ij \qquad (4.6.6)$$

$$\frac{\partial^2 g_\Delta{}^i}{\partial A_J \partial X_K} = -\frac{1}{A_J{}^2 \Delta_i{}^*} \tilde{\tilde{\psi}}_{(j)Ki} \qquad (4.6.9)$$

$$\frac{\partial^2 g_x{}^i}{\partial A_j \partial X_K} = -\frac{1}{A_j{}^2} \bar{\Phi}{}^j_{iK} \qquad (4.6.13)$$

4. **Input Arguments:** NR,NC,A,SIG,ND,UP,LOW,NE,NX,NDN,S,NSE,NTYPE,NDC,NCOL, NDT,NDTNX,NBOU,XC,YC,AREA, - See Glossary.

W = Working Storage
MUS = Vector containing stress μ's (length = NSNL)
MUD = Vector containing displacement μ's (length = NDCNL)
MUX = Vector containing redundant μ's (length = NXNL)
IRS = Vector identifying the active stress constraints and row label.
IRD = Vector identifying the active displacement constraints and row label.
IRX = Vector identifying the active redundants and row label.
IRA = Vector identifying the unknown design variables and column label.
YI = Mises-Hencky criteria for the element.

5. **Output Arguments:** Each partial is summed into the ARAY analysis matrix by S4310A.

6. **Error Returns:** None

7. **Calling Sequence:** CALL S4621 (ARAY,NR,NC,MUS,MUD,MUX,A,SIG,YI,ND,UP, LOW,NE,NX,NDN,S,NSE,NTYPE,W,NDC,IRS,IRD,IRX,IRA,NCOL, NDT,NDTNX,NBOU,XC,YC,AREA)

8. **Input Tapes:** I13

9. **Output Tapes:** None

10. Scratch Tapes: None

11. Storage Required: (2312 bytes) 578 words

12. Subroutine User: S461

13. Subroutine Required: S466, S4310A

1. **Subroutine Name:** S4622

2. **Purpose:** Compute $\frac{\partial^2 L}{\partial X_j \partial X_K} = \mu_\sigma^{i*} \frac{\partial^2 g_\sigma^{i*}}{\partial X_j \partial X_K}$

3. **Equations and Procedures:**

$$\frac{\partial^2 g_\sigma^{\;i}}{\partial X_j \partial X_K} = \frac{1}{A_i \sigma_i^{\;*}} \frac{\partial^2 Y_i}{\partial S_i^{(\ell)} \partial S_i^{(m)}} b_1{}^{i}{}_{\ell K}\, b_1{}^{i}{}_{mj} \qquad (4.6.7)$$

4. **Input Arguments:** A,SIG,Y,S,NTYPE,AREA,XC,YC,NE,NSE,NDT,NX,IR,IC, NTRANS,NR,NC,NCOL,NDTNX,NROW,NBOU

W - Working storage
MUS - Vector containing stress μ's (length = NSNL)
IRS - Vector identifying this active μ's.

5. **Output Arguments:** Partial after multiplication by $\mu\sigma^i$ that corresponds to it and is active is summed into the ARAY analysis matrix by S4310A.

6. **Error Returns:** None

7. **Calling Sequence:** CALL 54622 (A,SIG,Y,S,NTYPE,AREA,XC,YC,NE,NSE,NDT, NX,IR,IC,NTRANS,ARAY,NR,NC,NCOL,NDTNX,NROW,NBOU,W, MUS,IRS)

8. **Input Tapes:** NTAPE=I12.

9. **Output Tapes:** None

10. **Scratch Tapes:** None

11. **Storage Required:** (3268 bytes) 817 words

12. **Subroutine User:** S461

13. **Subroutine Required:** S4310A,S4622A,TRPRNT

1.	Subroutine Name:	S4622A
2.	Purpose:	To compute components of $\frac{\partial^2 Yi}{\partial S_i^{(\ell)} \partial S_i^{(m)}}$ for triangles and quads.
3.	Equations and Procedures:	The bar and shear elements have a zero value for this quantity. The quantities output from this routine are used to form a 3 x 3 matrix in S4622 for the triangular element. The quad element uses AA to FF with some modification to form a 5 x 5 matrix in S4622.
4.	Input Arguments:	S1,S2,S3,A,Y - Mises Hinckey criteria. AREA
5.	Output Arguments:	AA,BB,CC,DD,EE,FF - components of stress
6.	Error Returns:	None
7.	Calling Sequence:	CALL S4622A (S1,S2,S3,A,Y,AA,BB,CC,DD,EE,FF)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(672 bytes) 168 words
12.	Subroutine User:	S4622
13.	Subroutine Required:	None

1. **Subroutine Name:** S466

2. **Purpose:** Calculate 2nd partial of $g\sigma$ with respect to A_j and X_K.

3. **Equations and Procedures:**

$$\frac{\partial^2 g\sigma^i}{\partial A_j \partial X_K} = - \frac{1}{A_j^2 \sigma_i^*} \frac{\partial Yi}{\partial S_i^{(\ell)}} b_1{}^i{}_{\ell k} \delta ij \qquad (4.6.6)$$

4. **Input Arguments:** A,SIG,Y,S,NTYPE,AREA,XC,YC,NE,NSE,NDT,NX,IR,IC, NTRANS,NR,NC,NCOL,NDTNX,NROW,NBOU, - See Glossary.

   W - Working Storage
   MUS - Vector containing stress μ's (length - NSNL)
   IRS - Vector identifying the active μ's.

5. **Output Arguments:** The element of the partial multiplied by the corresponding $\mu\sigma$ is put into ARAY the analysis matrix.

6. **Error Returns:** None

7. **Calling Sequence:** CALL S466 (A,SIG,Y,S,NTYPE,AREA,XC,YC,NE,NSE,NDT, NX,IR,IC,NTRANS,ARAY,NR,NC,NCOL,NDTNX,NROW,NBOU,W, MUS,IRS)

8. **Input Tapes:** I12

9. **Output Tapes:** None

10. **Scratch Tapes:** None

11. **Storage Required:** (2698 bytes) 677 words

12. **Subroutine User:** S4621,S4621A

13. **Subroutine Required:** S4310A,TRPRNT

1.	Subroutine Name:	TAPE11
2.	Purpose:	Generate files I11,and structure cutter initialization
3.	Equations and Procedures:	File NSS1 (unit 1) is read and the structure cutter matrix is generated. This matrix is processed by routine AA. File I11 (General Problem Information) is written for further processing in the program. Routine S241 is called to generate the basic normalized structural matrices ϕ^{-1}, ψ and Ω.
4.	Input Arguments:	All calling arguments set up working storage for this routine and its subroutines.
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	CALL TAPE11 (A,A7,A8,KL,IELT,SIGU,DISPU,DISPL,AREA, ALL,DELTA,ALL2,A1,A2,A3,A4,A5,A6,WT,XC,YC,W,BETA,WA,NR,NC)
8.	Input Tapes:	NSS1
9.	Output Tapes:	I11,
10.	Scratch Tapes:	None
11.	Storage Required:	(5396 bytes) 1349 words
12.	Subroutine User:	LINK2
13.	Subroutine Required:	S241,MRPRNT,AA

1.	Subroutine Name:	TESTR
2.	Purpose:	Test for divergence in design.
3.	Equations and Procedures:	a) Compute the ratio of (new design/old design). b) Compare ratio to a criteria and set error switch if ratio is greater than criteria or less than 1/criteria.
4.	Input Arguments:	A,G,W,IAREA,NE,NXNL,PRI - See Glossary for definitions.
5.	Output Arguments:	IERR
6.	Error Returns:	IERR
7.	Calling Sequence:	CALL TESTR (A,G,W,IAREA,NE,NXNL,IERR,PRI)
8.	Input Tapes:	None
9.	Output Tapes:	None
10.	Scratch Tapes:	None
11.	Storage Required:	(964 bytes) 241 words
12.	Subroutine User:	S451,S461
13.	Subroutine Required:	None

1.	Subroutine Name:	TRPRNT
2.	Purpose:	Transpose matrix print.
3.	Equations and Procedures:	Print transpose of a matrix.
4.	Input Arguments:	NR - Number of rows in transposed matrix. NC - Number of columns in transposed matrix. AMTRIX - Matrix to print
5.	Output Arguments:	None
6.	Error Returns:	None
7.	Calling Sequence:	CALL TRPRNT (NR,NC,AMTRIX)
8.	Input Tapes:	None
9.	Output Tapes:	I01, Unit 10
10.	Scratch Tapes:	None
11.	Storage Required:	(490 bytes) 123 words
12.	Subroutine User:	FSD,S4622,S241,S231
13.	Subroutine Required:	None

A.5 PROBLEM SIZE LIMITATIONS

The key limitations are brought about by the total number of elements, number of load conditions and the number of degrees of freedom in the problem.

Due to dynamic storage allocation techniques the problem size can be controlled by changing two cards in the 'MAIN' routine. Both the dimension of the 'WORK' array and the size of the variable 'NWORK' must be equal. The delivered size is 20,100 words, but may be adjusted to your system.

If there is insufficient storage space defined for a problem the program will print a message indicating the amount of storage required for the problem and the amount of storage reserved by the above mentioned cards in the 'MAIN' routine. To execute the problem modify the dimension and 'NWORK' variable to be what the problem needs or reduce the number of elements, number of load conditions and/or number of degrees of freedom indicated on the input sheet.

APPENDIX B

RAPID REANALYSIS COMPUTER CODE

S. Gellin

TABLE OF CONTENTS

LIST OF FIGURES

B.1 INTRODUCTORY REMARKS

An interactive FORTRAN computer code was written to perform static and dynamic analyses of damaged structures using the rapid reanalysis theory, procedures and terminology discussed in Section 2.4 of Volume 1, Ref. 1. Structural idealizations are limited to the use of bar elements. Both large scale damage ($d_k^i=1$, Type A) and small scale damage ($d_k^i<1$, Type B) cases are included in the computer code.

B.2 PROGRAM CAPABILITIES AND LIMITATIONS

The computer program described herein performs basic static and dynamic analyses and rapid reanalyses for damage types A and B on plane truss structures. These can either be subjected to a given applied load or be in a state of free vibration, but not both simultaneously. Several damage cases of either type may be considered within a given analysis case.

The maximum number of elements is limited to 50. The maximum number of grid points is 25, thus, the maximum number of degrees of freedom is 50. In addition, the maximum number of reaction forces is 10 and the maximum number of applied loads (global X and Y directions only) is 10. For a given structure, the maximum number of redundants (which equals the number of elements plus the number of reactions minus the number of degrees of freedom) is 20. The minimum number of redundants is 1; thus, truss structures that are statically determinate cannot be analyzed.

Roller supports are assumed to act in one of the coordinate directions. Caution is advised when performing a type A damage analysis, as the sophisticated grid point removal scheme as discussed in Volume 1 is not included in the computer program. The degrees of freedom are numbered according to grid point number and direction. Thus, grid point j has d.o.f 2j-1 in the X-direction and 2j in the Y-direction.

B.3 DESCRIPTION OF INPUT DATA

The so-called "free format" is used to input data into the program. Each "card" or "line" has several numbers separated by commas. A detailed description follows. [Bracketed paragraphs are only needed if a dynamics case is being run.]

Group	Number of Cards	Quantities Read
1	1	Number of grid points, number of elements, number of applied loads.

[Comment: If a dynamics analysis is desired, specify the number of applied loads as zero.]

Group	Number of Cards	Quantities Read
[2	1	Number of desired modes and frequencies, maximum number of iterations used in eigenvalue routine.]
3	Number of grid points, in numerical order.	X-coordinate of grid point, Y-coordinate of grid point, support code.

Comment: Support code = 0 for unsupported grid point; = 1 for constrained in the X-direction only; = 2 for constrained in the Y-direction only; = 3 for pinned condition.

Group	Number of Cards	Quantities Read
4	Number of elements, in numerical order.	First grid point number of element, second grid point number of element.

Comment: "First" and "second" grid point numbers are in no particular order, i.e., they may be reversed.

Group	Number of Cards	Quantities Read
5	See comment below.	First element number, last element number, cross sectional area.

Comment: The form of this "card" helps to specify element properties in an efficient manner, particularly if most of the elements have the same value for the particular property. As an example, suppose, for a 12-bar truss, that all the elements have a value for a given property of 1., except element 8, which has a value for the property of 2. A series of cards which would specify this circumstance are:

1, 7, 1.
8, 8, 2.
9, 12, 1.

The card assigning the last element to its given value must be the last in the list. The others may be rearranged in any order.

Group	Number of Cards	Quantities Read
6	See comment after 5 above.	First element number, last element number, Young's modulus.

Group	Number of Cards	Quantities Read
[7	See Comment after 5 above.	First element number, last element number, mass density.]

[Skip group 8 and proceed to group 9.]

8	Number of applied loads.	Degree of freedom number, value of applied load.

Comment: The sign of the value determines the sense of the load for the specified degree of freedom.

9	1	Number of damage cases.

Comments: If the number of damage cases is specified as zero, then the program ends after completing a basic static or dynamic analysis. No further input cards would be needed.

10	Number of damage cases.	See comment below.

Comment: What follows is a description of several subgroupings of input data listed as 10.1, 10.2, etc. The collection of these subgroups form the group type 10. The number of group 10 type units is the number of damage cases.

10.1	1	Damage type, number of damaged elements.

Comment: For type B damage, set type = 1; for type A damage, set type = 2.

If damage type A is specified, proceed to subgroup 10.3.

10.2	Number of damaged elements.	Number of an element that has sustained damage, stiffness damage level.

[Proceed to subgroup 10.4.]

End of group type.

10.3	Number of damaged elements.	Number of an element that has sustained damage.

[Proceed to subgroup 10.4.]

End of group type.

[10.4	Number of damaged elements.	Mass damage level.]

[Comment: Order cards in reference to the order of elements presented in subgroups 10.2 or 10.3.]

[End of group type.]

End of input.

B.4 DESCRIPTION OF OUTPUT DATA

The results of the basic analysis, whether static or dynamic, is printed first, followed by the results of the various damage cases stipulated.

For static analysis only, three classes of output are given and labeled as "ELEMENT FORCES", "REACTION FORCES", and "DISPLACEMENTS".

"ELEMENT FORCES"	Number of elements.	Element number, corresponding force.

Comment: Positive quantities imply tension, negative quantities imply compression.

"REACTION FORCES"	Number of reactions.	Degree of freedom number that reaction corresponds to, reaction force.

Comment: Reactions positive in negative directions.

"DISPLACEMENTS	Number of degrees of freedom.	Degree of freedom number, displacement.

For dynamic analyses the above headings and outputs are still used, but they are taken to mean force, reaction and displacement mode shapes, respectively. The three sets are preceded by the frequency corresponding to the mode shape, and a set of these four subsets are presented for the number of modes specified in group 2 of the input.

If a set of output data is presented for a damage case, the set is preceded by a title specifying whether it is for Type A or B damage. In addition, there are diagnostic messages to indicate trouble spots; for instance, if the convergence criteria for a given calculation was not met in a certain prescribed number of calculations.

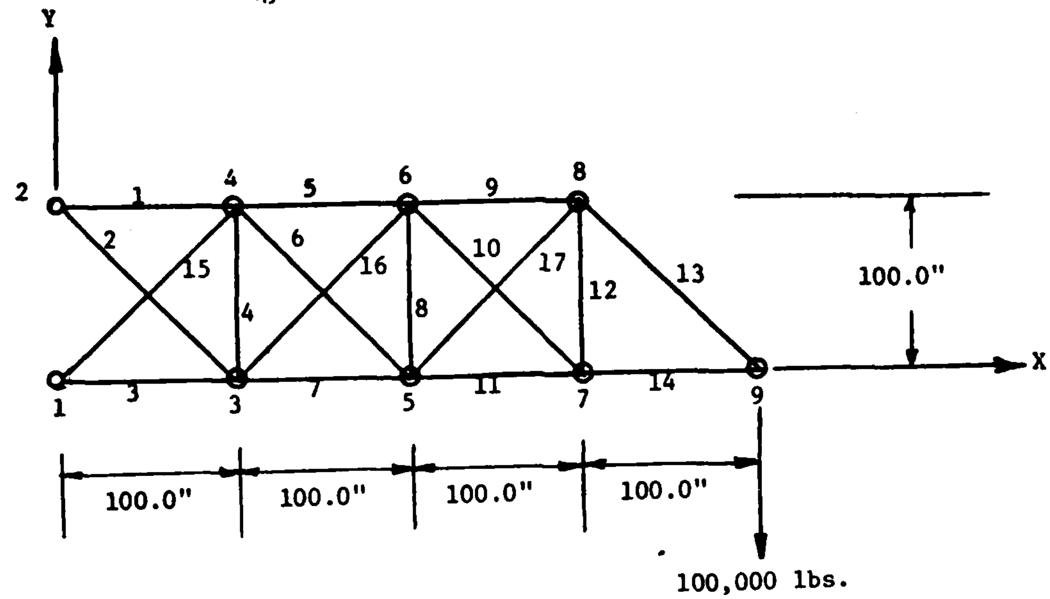
B.5 ILLUSTRATIVE EXAMPLE

A seventeen bar truss was used to demonstrate program capabilities. Figure B.1 depicts the truss and lists the damage cases. Cases D1-2 are the dynamic counterparts to static cases S1-2. The input data for the statics problems is shown in Figure B.2, and the analysis results in Figure B.3. The input for the dynamics problems is shown in Figure B.4, and the output in Figure B.5.

The estimated compile time for this program is about 6.2 seconds on the IBM 3031 computer using the VSPC system; the estimated execution time is .2 seconds per analysis (i.e., 3 mode shape determinations is counted as 3 analyses).

A program listing is given in Figure B.6.

Figure B.1 Seventeen Bar Truss Example



$E = 30 \times 10^6$ psi
$A = 5$ in^2
$\rho = .0083$ slugs/in^3

for all members

Damage Cases

Case	Damaged Elements	Stiffness Damage d^i_k	Mass Damage d^i_m
S1	15,16,17	.1 all 3	NA
S2	15	1.0	NA
D1	15,16,17	.1 all 3	0 all 3
D2	15	1.0	1.0

Figure B.2 Input for Statics Cases

```
10 9,17,1
20 0.,0.,3
30 0.,100.,3
40 100.,0.,0
50 100.,100.,0
60 200.,0.,0
70 200.,100.,0
80 300.,0.,0
90 300.,100.,0
100 400.,0.,0
110 2,4
120 2,3
130 1,3
140 3,4
150 4,6
160 4,5
170 3,5
180 5,6
190 6,8
200 6,7
210 5,7
220 7,8
230 8,9
240 7,9
250 1,4
260 3,6
270 5,8
280 1,17,5.
290 1,17,3.+7
300 18,-100000.
310 5
320 1,3
330 15,.1
340 16,.1
350 17,.1
360 1,3
370 15,.5
380 16,.5
390 17,.5
400 2,1
410 15
420 2,1
430 16
440 2,1
450 17
```

Figure B.3 Output for Statics Cases

Figure B.3.1 Basic Analysis

ELEMENT FORCES

1	0.3501E+06
2	0.7062E+05
3	-0.3499E+06
4	-0.4852E+03
5	0.2495E+06
6	0.7149E+05
7	-0.2505E+06
8	0.4686E+04
9	0.1552E+06
10	0.6331E+05
11	-0.1448E+06
12	-0.4477E+05
13	0.1414E+06
14	-0.1000E+06
15	-0.7080E+05
16	-0.6993E+05
17	-0.7811E+05

REACTION FORCES

1	-0.4000E+06
2	-0.5006E+05
3	0.4000E+06
4	-0.4994E+05

DISPLACEMENTS

1	0.0
2	0.0
3	0.0
4	0.0
5	-0.2333E+00
6	-0.3275E+00
7	0.2334E+00
8	-0.3278E+00
9	-0.4003E+00
10	-0.1057E+01
11	0.3997E+00
12	-0.1054E+01
13	-0.4968E+00
14	-0.2035E+01
15	0.5032E+00
16	-0.2064E+01
17	-0.5635E+00
18	-0.3320E+01

Figure B.3 (Continued)

Figure B.3.2 Damage Case S1

TYPE B DAMAGE

ELEMENT FORCES

1	0.3484E+06
2	0.7291E+05
3	-0.3516E+06
4	-0.3336E+04
5	0.2482E+06
6	0.7323E+05
7	-0.2518E+06
8	0.1835E+04
9	0.1536E+06
10	0.6559E+05
11	-0.1464E+06
12	-0.4638E+05
13	0.1414E+06
14	-0.1000E+06
15	-0.6851E+05
16	-0.6819E+05
17	-0.7583E+05

REACTION FORCES

1	-0.4000E+06
2	-0.4845E+05
3	0.4000E+06
4	-0.5155E+05

DISPLACEMENTS

1	0.0
2	0.0
3	0.0
4	0.0
5	-0.2344E+00
6	-0.3316E+00
7	0.2323E+00
8	-0.3338E+00
9	-0.4022E+00
10	-0.1066E+01
11	0.3978E+00
12	-0.1065E+01
13	-0.4998E+00
14	-0.2050E+01
15	0.5002E+00
16	-0.2081E+01
17	-0.5665E+00
18	-0.3336E+01

Figure B.3 (Continued)

Figure B.3.4 Damage Case S2

TYPE A DAMAGE

ELEMENT FORCES

1	0.3000E+06
2	0.1414E+06
3	-0.4000E+06
4	-0.5585E+05
5	0.2441E+06
6	0.7899E+05
7	-0.2559E+06
8	-0.1281E+05
9	0.1430E+06
10	0.8054E+05
11	-0.1570E+06
12	-0.5695E+05
13	0.1414E+06
14	-0.1000E+06
15	-0.6250E-01
16	-0.6243E+05
17	-0.6088E+05

REACTION FORCES

1	-0.4000E+06
2	-0.1250E+00
3	0.4000E+06
4	-0.1000E+06

DISPLACEMENTS

1	0.0
2	0.0
3	0.0
4	0.0
5	-0.2667E+00
6	-0.4552E+00
7	0.2000E+00
8	-0.4924E+00
9	-0.4372E+00
10	-0.1235E+01
11	0.3628E+00
12	-0.1243E+01
13	-0.5419E+00
14	-0.2255E+01
15	0.4581E+00
16	-0.2293E+01
17	-0.6085E+00
18	-0.3549E+01

Figure B.4 Input for Dynamics Cases

```
  10 9,17,0
  15 3,10
  20 0.,0.,3
  30 0.,100.,3
  40 100.,0.,0
  50 100.,100.,0
  60 200.,0.,0
  70 200.,100.,0
  80 300.,0.,0
  90 300.,100.,0
 100 400.,0.,0
 110 2,4
 120 2,3
 130 1,3
 140 3,4
 150 4,6
 160 4,5
 170 3,5
 180 5,6
 190 6,8
 200 6,7
 210 5,7
 220 7,8
 230 8,9
 240 7,9
 250 1,4
 260 3,6
 270 5,8
 280 1,17,5.
 290 1,17,3.+7
 300 1,17,.0083
 310 5
 320 1,3
 330 15,.1
 340 16,.1
 350 17,.1
 351 0.
 352 0.
 353 0.
 360 1,3
 370 15,.5
 380 16,.5
 390 17,.5
 391 0.
 392 0.
 393 0.
 400 2,1
 410 15
 411 1.
 420 2,1
 430 16
 431 1.
 440 2,1
 450 17
 451 1.
```

Figure B.5 Output for Dynamics Case

Figure B.5.1 Basic Analysis

```
CONVERGENCE PROBLEMS IN POWER - Z =  0.3332E-01

FREQUENCY( 1) =    0.4156E+02

ELEMENT FORCES

        1       -0.1601E+02
        2       -0.4930E+01
        3        0.1614E+02
        4       -0.2319E-01
        5       -0.9135E+01
        6       -0.4434E+01
        7        0.9278E+01
        8       -0.8789E-01
        9       -0.3641E+01
       10       -0.2825E+01
       11        0.3627E+01
       12        0.5629E+00
       13       -0.1623E+01
       14        0.1323E+01
       15        0.4979E+01
       16        0.4450E+01
       17        0.3056E+01

REACTION FORCES

        1        0.1966E+02
        2        0.3521E+01
        3       -0.1950E+02
        4        0.3486E+01

DISPLACEMENTS

        1        0.0
        2        0.0
        3        0.0
        4        0.0
        5        0.1076E-04
        6        0.1734E-04
        7       -0.1067E-04
        8        0.1732E-04
        9        0.1695E-04
       10        0.5085E-04
       11       -0.1675E-04
       12        0.5079E-04
       13        0.1937E-04
       14        0.9068E-04
       15       -0.1917E-04
       16        0.9105E-04
       17        0.2025E-04
       18        0.1326E-03
```

Figure B.5 (Continued)

Figure B.5.1 (Continued)

```
FREQUENCY( 2) =     0.1570E+03

ELEMENT FORCES

        1         -0.2573E+01
        2         -0.3114E+01
        3         -0.1557E+01
        4          0.1007E+01
        5          0.1614E+00
        6         -0.1528E+01
        7         -0.3779E+01
        8          0.6802E+00
        9          0.1062E+01
       10          0.1565E+00
       11         -0.3087E+01
       12         -0.1461E+00
       13          0.1147E+01
       14         -0.1662E+01
       15          0.1553E+01
       16          0.5628E+00
       17         -0.9023E+00

REACTION FORCES

        1         -0.4586E+00
        2          0.1098E+01
        3         -0.4775E+01
        4          0.2202E+01

DISPLACEMENTS

        1          0.0
        2          0.0
        3          0.0
        4          0.0
        5         -0.1267E-05
        6          0.2675E-05
        7         -0.1865E-05
        8          0.3440E-05
        9         -0.3808E-05
       10          0.3485E-05
       11         -0.2058E-05
       12          0.3990E-05
       13         -0.5826E-05
       14          0.1572E-06
       15         -0.1569E-05
       16          0.1091E-06
       17         -0.6897E-05
       18         -0.6573E-05
```

Figure B.5 (Continued)

Figure B.5.2 Damage Case D1

```
TYPE B DAMAGE
CONVERGENCE PROBLEMS IN POWER - Z =  0.2834E-02

FREQUENCY( 1) =    0.4137E+02

ELEMENT FORCES

    1        -0.1590E+02
    2        -0.5100E+01
    3         0.1626E+02
    4         0.1718E+00
    5        -0.9041E+01
    6        -0.4553E+01
    7         0.9356E+01
    8         0.5329E-01
    9        -0.3571E+01
   10        -0.2908E+01
   11         0.3682E+01
   12         0.6212E+00
   13        -0.1619E+01
   14         0.1320E+01
   15         0.4830E+01
   16         0.4341E+01
   17         0.2971E+01

REACTION FORCES

    1         0.1967E+02
    2         0.3415E+01
    3        -0.1950E+02
    4         0.3606E+01

DISPLACEMENTS

    1         0.0
    2         0.0
    3         0.0
    4         0.0
    5         0.1084E-04
    6         0.1764E-04
    7        -0.1060E-04
    8         0.1775E-04
    9         0.1707E-04
   10         0.5149E-04
   11        -0.1662E-04
   12         0.5153E-04
   13         0.1953E-04
   14         0.9156E-04
   15        -0.1901E-04
   16         0.9197E-04
   17         0.2041E-04
   18         0.1335E-03
```

Figure B.5 (Continued)

Figure B.5.2 (Continued)

FREQUENCY(2) = 0.1495E+03

ELEMENT FORCES

1	-0.1138E+01
2	-0.3319E+01
3	-0.3623E-01
4	0.4062E+00
5	0.1953E+01
6	-0.1433E+01
7	-0.3125E+01
8	0.2758E+00
9	0.2144E+01
10	0.6058E+00
11	-0.2886E+01
12	-0.4320E+00
13	0.1475E+01
14	-0.1622E+01
15	0.2641E+01
16	0.1069E+01
17	-0.9493E+00

REACTION FORCES

1	0.1831E+01
2	0.1867E+01
3	-0.3485E+01
4	0.2347E+01

DISPLACEMENTS

1	0.0
2	0.0
3	0.0
4	0.0
5	-0.4879E-07
6	0.4382E-05
7	-0.7788E-06
8	0.4663E-05
9	-0.2149E-05
10	0.5210E-05
11	0.5018E-06
12	0.5400E-05
13	-0.4082E-05
14	0.1757E-07
15	0.1919E-05
16	-0.2674E-06
17	-0.5167E-05
18	-0.9317E-05

Figure B.5 (Continued)

Figure B.5.4 Damage D2

```
TYPE A DAMAGE

FREQUENCY( 1) =    0.3922E+02

ELEMENT FORCES

   1      -0.1215E+02
   2      -0.1027E+02
   3       0.1985E+02
   4       0.3187E+01
   5      -0.9259E+01
   6      -0.3928E+01
   7       0.8771E+01
   8      -0.4460E+00
   9      -0.3429E+01
  10      -0.2814E+01
  11       0.3558E+01
  12       0.5715E+00
  13      -0.1547E+01
  14       0.1266E+01
  15       0.9537E-05
  16       0.5054E+01
  17       0.2941E+01

REACTION FORCES

   1       0.1985E+02
   2       0.1049E-04
   3      -0.1941E+02
   4       0.7262E+01

DISPLACEMENTS

   1       0.0
   2       0.0
   3       0.0
   4       0.0
   5       0.1323E-04
   6       0.2692E-04
   7      -0.8102E-05
   8       0.2905E-04
   9       0.1908E-04
  10       0.6147E-04
  11      -0.1428E-04
  12       0.6117E-04
  13       0.2146E-04
  14       0.1007E-03
  15      -0.1656E-04
  16       0.1010E-03
  17       0.2230E-04
  18       0.1420E-03
```

Figure B.5 (Continued)

Figure B.5.4 (Continued)

FREQUENCY(2) = 0.1224E+03

ELEMENT FORCES

1	0.3593E+01
2	-0.5343E+01
3	0.1976E+01
4	0.1458E+01
5	0.3434E+01
6	-0.2397E+00
7	-0.2833E+01
8	-0.2924E+00
9	0.2699E+01
10	0.9633E+00
11	-0.2139E+01
12	-0.4803E+00
13	0.1370E+01
14	-0.1180E+01
15	0.9537E-05
16	0.1123E+01
17	-0.1070E+01

REACTION FORCES

1	0.1976E+01
2	0.8404E-05
3	-0.1849E+00
4	0.3778E+01

DISPLACEMENTS

1	0.0
2	0.0
3	0.0
4	0.0
5	0.1304E-05
6	0.8422E-05
7	0.2396E-05
8	0.9395E-05
9	-0.5929E-06
10	0.6725E-05
11	0.4687E-05
12	0.6532E-05
13	-0.2024E-05
14	-0.1464E-05
15	0.6487E-05
16	-0.1784E-05
17	-0.2812E-05
18	-0.1291E-04

LIST OF REFERENCES

(1) J. R. Batt, S. Gellin, and R. A. Gellatly, "Force Method Optimization II, Volume I", Theoretical Development, AFWAL-TR-82-3088, December 1982.

(2) R. A. Gellatly and R. D. Thom, "Force Method Optimization", AFWAL-TR-80-3006, February 1980.